Success and Failure Factors in ICT Projects: A Dutch Perspective

Success and Failure Factors in ICT Projects: A Dutch Perspective

Aart J. van Dijk MSc EMITA
Master of Science in Engineering (ICT)
Executive Master of IT Auditing

Thesis submitted as partial fulfilment of the Engineering Doctorate (EngD) programme

School of Engineering and Information Sciences

Middlesex University

This thesis was approved on 1 May 2009

Exam Board

External examiner:	Professor Margaret Ross MBE
Internal examiner:	Dr Chris Sadler
Chair:	Dr Mehmet Karamanoglu
Director of Studies:	Professor Richard Comley
Supervisor:	Professor Anthony White
Local supervisor:	Dr Paul Mantelaers (Netherlands Court of Audit)

Doctor of Engineering
School of Engineering and Information Sciences
Middlesex University, Hendon, London

I hereby declare that this thesis embodies the result of the course of my study and research. It has been composed by myself and has not been presented for a higher degree or any other award to any other institution.

Student's Signature:

Student's Name: Aart J. van Dijk

Date of Submission: May 2009

To **Janny**

who over the period 1965-2009 provided me with the space to spend a substantial part of my leisure time studying, teaching and publishing

Publisher: Ridderprint b.v.
Postbus 46
2980 AA Ridderkerk
The Netherlands
Tel: (+31) (0)180 463962

By order of: Avédé-Info b.v.
Information Science Consultancy and IT Auditing
Ponyweide 79A
2727 HK Zoetermeer
The Netherlands
Tel: (+31) (0)79 3417194
E-mail: avede-info@planet.nl

Printed by: Ridderprint b.v.

CIP-DATA KONINKLIJKE BIBLIOTHEEK, THE HAGUE

Dijk, Aart J. van

Success and Failure Factors in ICT Projects: A Dutch Perspective /
Aart J. van Dijk
EngD thesis Middlesex University London – References included
ISBN/EAN: 978-90-5335-200-7
Keywords: Success/failure factors, SUFFI, Dutch ICT projects, Software Project Dynamics, project management, software audits

Acknowledgements

I was fortunate to carry out this study and research under supervision of Professor Anthony White and Professor Richard Comley of the School of Engineering and Information Sciences of the Middlesex University in London. It concerns an "Engineering Doctorate (EngD)". Characteristics of an EngD are amongst other things:

- EngD is a four-year postgraduate research award intended for leading research engineers who want a managerial career;
- the work is project based (or portfolio based) and has to make a significant contribution to the performance of the company/companies;
- the EngD provides a more vocationally oriented doctorate in engineering;
- the EngD is fully equivalent in academic status to a traditional PhD.

Over the last 44 years, I have been intensively involved in the design, building, implementation, management and auditing of information systems. I often did this in the role of (working) project manager or as IT auditor. The "portfolio of projects" that also comes up for discussion in this thesis, consists of an important part of my projects and project audits of the past 44 years.

With an EngD there should be a "sponsoring business" and a local supervisor. To my obvious delight, the Netherlands Court of Audit (Algemene Rekenkamer) in The Hague was willing to act as a "sponsoring business".

First, I would like to thank the Middlesex University in London and in particular my supervisors, Professor Anthony White and Professor Richard Comley for their efforts in offering me the opportunity to base my studies for the EngD on my portfolio of projects. I would like to thank Professor Anthony White for his commitment, supervision and prompt answers to my queries during the study, for which almost all communication was conducted by means of the Internet. Furthermore, I would like to thank the Netherlands Court of Audit for the trust they have shown in me and for their cooperation. Dr Paul Mantelaers of the Netherlands Court of Audit was more than just my critical local supervisor but was sometimes my help and my rock. Alice Saunders took care of translation of a large part of my texts into professional English.

Finally, I would like to extend cordial thanks to all colleagues, users and customers who were in some way involved in my projects or audits over the last 44 years. Without them, there would never have been a portfolio of projects and no EngD either.

Aart J. van Dijk, Zoetermeer, May 2009

TABLE OF CONTENTS

		Page
ABSTRACT		
1	**PROBLEM DEFINITION**	**1**
1.1	INTRODUCTION	1
1.2	THE RESEARCH QUESTION	2
1.3	EXPLANATION CONCEPTS	3
1.3.1	What others say happens	3
1.3.2	Portfolio of projects	4
1.4	PROBLEM DETAILS	5
1.5	APPROACH TO AND REPORT OF THE STUDY	5
2	**WHAT IS UNDERSTOOD BY A PROJECT FAILURE?**	**9**
2.1	DEFINITIONS	9
2.2	TAXONOMY OF PROBLEM PROJECTS	9
2.3	TROUBLED/FAILED PROJECTS	12
2.3.1	Definition of a troubled project	12
2.3.2	Definition of a project failure	13
2.4	DEFINITION OF THE PROBLEM [Answer to sub question 1]	13
3	**WHAT DO OTHERS THINK OF ICT PROJECTS FAILING OR NOT FAILING?**	**15**
3.1	INTRODUCTION	15
3.2	THE PROJECT LIFE-CYCLE AND ROOT CAUSES OF TROUBLED PROJECTS	16
3.2.1	Introduction	16
3.2.2	Government ICT projects	18
3.2.3	Benefits	19
3.3	THE SIZE OF PROJECTS	19
3.3.1	Introduction	19
3.3.2	Size of project	20
3.3.3	Six sub industries	22
3.3.4	Conclusions	22
3.4	PROJECT RISKS BASED	23
3.4.1	Introduction	23
3.4.2	Characteristics	23
3.4.3	Risk designators	23
3.4.4	Characteristics and their risk designators	25
3.4.5	Analysis and conclusions	26
3.5	OTHER AUTHORS	28
3.5.1	Introduction	28
3.5.2	Yeo (2002)	28
3.5.3	May (1998)	29
3.5.4	Reel (1999)	30
3.6	THE DUTCH SITUATION	31
3.6.1	Introduction	31
3.6.2	The past	32
3.6.2.1	*Introduction*	32
3.6.2.2	*The subjects discussed by Ooonincx*	33
3.6.3	The opinion of the ICT lawyers	33
3.6.3.1	*Introduction*	33

3.6.3.2	*The problem of failing projects*	34
3.6.3.3	*The field study*	35
3.6.4	The present	39
3.6.4.1	*Introduction*	39
3.6.4.2	*Failure factors*	40
3.6.4.3	*Study design*	40
3.6.4.4	*Results*	40
3.6.5	The methods	46
3.6.5.1	*Introduction*	46
3.6.5.2	*Methods used*	47
3.6.5.3	*Project manegement is risk manegement*	48
3.7	COHERENCE	49
3.7.1	Introduction	49
3.7.2	Results	50
3.8	SOFTWARE PROJECT DYNAMICS – AN INTEGRATED APPROACH	62
3.8.1	Introduction	62
3.8.2	SUFFI table TAH	62
3.8.3	Remarks	68
3.8.4	SUFFI tables TAH A and B	71
3.8.5	Conclusions	75
3.8.6	The SUFFI model	77
4	**WHICH ROOT CAUSES ARE THE 'BIG HITTERS'?**	79
4.1	VIEW OF THE AUTHORS	79
4.2	SUMMARY	83
4.3	DEFINITION OF THE PROBLEM [Answers to sub questions 2 and 3]	84
5	**THE PORTFOLIO OF PROJECTS**	**85**
5.1	INTRODUCTION	85
5.2	CASE: POTVIS (KLPD)	94
5.2.1	Introduction	94
5.2.2	Outline article RBAC (2004)	95
5.2.3	Management aspects	99
5.3	CASE: KOLIBRIE (KPN TELECOM)	104
5.3.1	Introduction	104
5.3.2	Outline Kolibrie articles	104
5.3.2.1	*Data exchange (interfacing) of applications using a message broker (2001)*	104
5.3.2.2	*EAI and ERP with Baan Open World (2001)*	107
5.3.3	Management aspects	109
5.4	CASE: CHARGING METHOD (GAK)	113
5.4.1	Introduction	113
5.4.2	Outline article Charging method (1998)	113
5.4.3	Management aspects	118
5.5	CASE: TELEPHONY (DUT)	121
5.5.1	Introduction	121
5.5.2	Outline article IMPALA (1994)	121
5.5.3	Management aspects	123

5.6	CASE: OKAPI (UoA)	129
5.6.1	Introduction	129
5.6.2	Outline OKAPI article (1994)	129
5.6.3	Management aspects	130
5.7	CASE: GIRAF (DUT)	134
5.7.1	Introduction	134
5.7.2	Outline GIRAF articles	134
5.7.2.1	*GIRAF (1984)*	134
5.7.2.2	*ADS/ONLINE (1984)*	135
5.7.3	Management aspects	136
5.8	CASE: AUBID (DUT)	140
5.8.1	Introduction	140
5.8.2	Outline AUBID articles	140
5.8.2.1	*Delft students request library books from computer (1979)*	140
5.8.2.2	*Experiences with the CICS/VS Command-level (1980)*	141
5.8.3	Management aspects	143
5.9	CASE: VDV (DUT)	148
5.9.1	Introduction	148
5.9.2	Outline article VDV (1975)	148
5.9.3	Management aspects	151
5.10	CASE: BIBLIOSYSTEM (DUT)	154
5.10.1	Introduction	154
5.10.2	Outline BIBLIOSYSTEM article (1971)	154
5.10.3	Management aspects	155
5.11	PROJECT AUDITS	157
5.11.1	Case: Multihouse versus Nutsbedrijven (1997)	157
5.11.1.1	*Introduction*	157
5.11.1.2	*A few points from the CDD report*	158
5.11.1.3	*Methods*	159
5.11.2	Case: SYSA (GOVERN) (2004)	161
5.11.2.1	*Introduction*	161
5.11.2.2	*Investigation*	161
5.11.2.3	*Conclusions and recommendations*	162
5.11.2.4	*Reaction*	164
5.11.2.5	*Methods*	165
5.11.3	Case: ACCINT (PUBLIC) (2004)	167
5.11.3.1	*Introduction*	167
5.11.3.2	*Main conclusions, recommendations and points open to improvement*	167
5.11.3.3	*Reactions*	170
5.11.3.4	*Methods*	170
5.11.4	Case: SOX (FINANCE) (2006)	173
5.11.4.1	*Introduction*	173
5.11.4.2	*SOX and SOX 404*	173
5.11.4.3	*Results*	174
5.11.4.4	*Methods*	175
5.11.4.5	*The Sóxima model*	178
5.12	DEFINITION OF THE PROBLEM [Answer to sub question 4]	178

6	**BIG HITTERS / SUCCESSFUL AND NOT SUCCESSFUL**	**179**
6.1	INTRODUCTION	179
6.2	CONCLUSIONS	179
6.3	DEFINITION OF THE PROBLEM [Answer to sub question 5]	185
7	**SUFFI – CHARTS**	**187**
7.1	INTRODUCTION	187
7.2	SUCCESS AND FAILURE FACTORS IN ICT PROJECTS: SUFFI CHART	188
7.3	SUCCESS AND FAILURE FACTORS IN ICT PROJECTS: SUFFI TOTAL CHART	194
7.4	DEFINITION OF THE PROBLEM [Answer to sub question 6]	203
8	**CASE "NETHERLANDS COURT OF AUDIT"**	**205**
8.1	INTRODUCTION	205
8.2	REPORT COURT OF AUDIT	205
8.3	TESTING OF THE CASE AGAINST THE SUFFI CHART	208
8.4	SUMMARY	213
8.4.1	Introduction	213
8.4.2	Conclusions	214
8.5	DEFINITION OF THE PROBLEM [Answer to sub question 7]	214
9	**DISCUSSION OF RESULTS**	**217**
9.1	INTRODUCTION	217
9.2	PORTFOLIO OF PROJECTS	217
9.3	BIG HITTER 1: POOR PROJECT MANAGEMENT	219
9.4	BIG HITTER 2: DEADLINES ARE UNREALISTIC	226
9.5	BIG HITTER 3: POOR COMMUNICATION	230
9.6	BIG HITTER 4: INCOMPLETE/ WEAK DEFINITION REQUIREMENTS	231
9.7	BIG HITTER 5: INSUFFICIENT INVOLVEMENT OF FUTURE USERS	234
9.8	BIG HITTERS: CONCLUSION	236
10	**CONCLUSIONS AND RECOMMENDATIONS**	**237**
10.1	INTRODUCTION	237
10.2	SUB QUESTIONS	237
10.2.1	Sub question 1	237
10.2.2	Sub question 2	238
10.2.3	Sub question 3	239
10.2.4	Sub question 4	239
10.2.5	Sub question 5	240
10.2.6	Sub question 6	240
10.2.7	Sub question 7	241
10.3	CONCLUSIONS	242
10.4	FURTHER RESEARCH	243
10.5	RECOMMENDATIONS	244
ABBREVIATIONS		**247**
REFERENCES		**251**

APPENDICES 255

APPENDIX 1: THE PUBLICATION OF JAN OONINCX [1982]: WHY ARE INFORMATION SYSTEMS STILL FAILING? 255
APPENDIX 2: THE PUBLICATION OF TAREK ABDEL-HAMID AND STUART E. MADNICK [1991]: SOFTWARE PROJECTS DYNAMICS – AN INTEGR. APPROACH 261
APPENDIX 3: THE SÓXIMA MODEL 283
APPENDIX 4: JOURNAL PUBLICATIONS 291

- Publication 1 293
- Publication 2 323
- Publication 3 331
- Publication 4 351
- Publication 5 373
- Publication 6 381
- Publication 7 401
- Publication 8 419
- Publication 9 431
- Publication 10 445
- Publication 11 463
- Publication 12 477
- Publication 13 489
- Publication 14 501

SUMMARY 511

CURRICULUM VITAE 519

ABSTRACT

This thesis examines the success and failure factors in ICT projects. The low success rate of software projects in terms of reliability, meeting due dates and working within assigned budgets is widely recognised and topical. International as well as Dutch publications and the procedures in Tarek Abdel-Hamid's work on Software Project Management/Dynamics are discussed. A SUFFI Chart (SUFFI = SUccess and Failure Factors in ICT projects) is developed. The management of a portfolio of projects is compared with the SUFFI Chart. A number of Dutch projects with which the author was directly involved are examined to show how they compare with the factors identified from the literature. These do show considerable correlation between important SUFFI factors and project success. The portfolio consists of nine ICT projects and four ICT project audits. Projects such as SAP, RBAC, EAI, charging method, PABX, financial building administration, information retrieval, book reservations, traffic data collection, introduction of the Internet functionality and SOX, for different companies/organisations (Delft University of Technology, National Police Services Agency, KPN – Dutch Telecom Company, University of Amsterdam, government, banking). This work shows that for a successful project 4 of the 5 most important SUFFIs have to be absent.

CHAPTER 1

PROBLEM DEFINITION

1.1 INTRODUCTION

On a regular basis, reports appear in the Press about ICT projects that fail and do not reach their original goal. This concerns both business as well as government projects. This often means significant losses with regard to investments. The failing of ICT projects is nothing new. The number of publications on this subject indicates that many researchers have been or are still studying this subject. As early as 1982, Professor Jan Oonincx wrote his book "Why are information systems still failing?" [Oonincx 1982]. His conclusions were: "*Information systems, which are set up too ambitiously, too isolated or without proper planning, stand a very large chance of failing. Insufficient involvement of future users in the development of information systems or a passive attitude of the top management also often lead to disappointing results*" [Van Dijk 1982].

In his inaugural speech at the Rijksuniversiteit Groningen in 2002, Professor Egon Berghout said [Berghout 2002] the following about Jan Oonincx' publication: "This booklet is, without a doubt, based on many years of annoyance preceding 1982, however twenty years later it can be reprinted almost unchanged. Hardly any causes have been removed".

The American "Standish Group" has been involved for 10 years in research into ICT [The Standish Group International 2003]. In their research, they aim, even more emphatically, at success and failure factors regarding ICT projects. Their study, which has been appropriately baptised "*Chaos*", appears every two years. This study also shows that in 2003 only 34% are successful, 51% does not go according to plan but ultimately does lead to some result and 15% of the projects fail completely.

"As far as ICT projects are concerned, the government handles these badly". The Netherlands Court of Audit arrived at this conclusion in the research report that appeared at the end of November 2007 [Wijsman et al. 2007]. The research report of the Netherlands Court of Audit confirmed that the Dutch government spends billions of Euros every single year on ICT projects that fail entirely or in part. The research was executed as a result of questions asked in the Dutch Lower Chamber. It is not clear how many projects and how much money are/is involved. The list of (partly) failed government ICT projects is a long one.

The abovementioned studies show that the subject is still very relevant in the year 2008 and it may be concluded that the subject of success and failure factors in ICT projects has been in the spotlight for more than 26 years and still is very topical.

1.2 THE RESEARCH QUESTION

This section defines the objectives and outlines the problem (the research question). Based on the definition of the problem, a number of sub questions are formulated [Kuypers 1982].

Objective (importance of knowledge of the study) (objective of)

(An academic exercise in) finding out (a contribution to) the true success and failure factors used in ICT practice (=scientific).

(A contribution to) obtaining an Engineering Doctorate (=social).

Definition of the problem

How were the ICT projects the author) worked on (the portfolio of projects) managed (the key here is the author's observations and experiences) with regard to success and failure factors, and how do they agree or disagree with what the procedures in Tarek Abdel-Hamid's work on Software Project Management and others say happens with regard to success and failure factors?*

*) in this chapter: the author is AvD

Sub questions

1. *What is understood by success/failure factors in ICT (for short: SUFFIs = SUccess and Failure Factors in ICT projects)?*

2. *Is it possible to derive SUFFIs from international publications, Dutch publications and from the procedures in Tarek Abdel-Hamid's work on Software Project Management and if so, what particular SUFFIs?*

3. *Are there any Big Hitters*) amongst the SUFFIs and if so what are these?*

4. *Which SUFFIs are applicable to what particular project from the portfolio of the author's projects?*

5. *Is it possible to use the Big Hitters for distinguishing between successful and not successful projects from the portfolio of the author's projects?*

6. *Is it possible to present the SUFFIs in a SUFFI Chart that is easy to use in practice by others?*

7. *Is it possible to apply the SUFFI Chart in the "Netherlands Court of Audit" case?*

*) 'Big Hitters' are the most important (most common) success and failure factors. John Smith introduced the name 'Big Hitter' [Smith 2001].

1.3 EXPLANATION CONCEPTS

1.3.1 What others say happens

In order to be able to establish how the projects the author worked on agree or disagree with what others say happens, it is necessary to map out a number of publications (what others say happens) in this field. To that purpose, the following publications will be discussed:

International publications

- Large Software System Failures and Successes [Jones 1996-1]
- Major Causes of Software Project Failures [May 1998]
- Critical Success Factors In Software Projects [Reel 1999]
- Seven Characteristics of Dysfunctional Software Projects [Evans et al. 2002]
- The 40 root causes of troubled IT projects [Smith 2002]
- Critical failure factors in information system projects [Yeo 2002]

Dutch publications

- Why are information systems still failing? [Oonincx 1982]
- Success and failure factors in complex ICT projects [Beenker 2004]
- ICT project management on the road to adulthood:
  Success factors for ICT projects [Noordam et al. 2007]

Tarek Abdel-Hamid

The procedures of Tarek Abdel-Hamid in:

"Software Projects Dynamics – An Integrated Approach" [Abdel-Hamid & Madnick 1991].

The writers of these publications report on research carried out in the field of success and failure factors of ICT projects. Some writers have also published books on this subject (for example Abdel-Hamid, Smith, Oonincx, Jones). For that reason, this collection of publications may also be considered representative for this subject as far as this thesis is concerned.

1.3.2 Portfolio of projects

The portfolio consists of 9 projects with external project-based publications and 4 (project-based) audits. Table 1.1 contains these projects and audits. About the projects the author has written 12 project-based publications in Dutch journals and 33 internal publications. The 9 projects represent an effort and duration of about 16 years. The author's role in projects 6-9 was: *internal* project manager at the Delft University of Technology. Regarding the projects 1-5, the author was the *external* project manager. The 9 projects on which external publications in the trade magazines appeared are discussed in sections 5.2 up to and including 5.10. In section 5.11, four (project-based) audits come up for discussion. The author was the *external* IT Auditor.

Remark in advance

Section 1.1, refers to research carried out by The American "Standish Group" [The Standish Group International 2003]. The author's experiences with projects within ICT over the last 43 years diverges from these research results. Not one of the author's projects was aborted prematurely. Some projects took more time/money than anticipated or were temporarily classed as "troubled project" but practically all projects were successfully implemented. Only the results of two projects the author was involved with have not been put into use:

- the CCIP project for the FORTIS BANK Netherlands. This project did produce the required result: an application (SCOPUS) was procured by FORTIS and was adjusted for FORTIS to be used for the support of operational ITIL processes. In the week, that the project result was delivered it was announced that the supplier of the SCOPUS application had been bought up by SIEBEL. For the FORTIS management, this was reason to put the project result on ice till further notice;
- during the author's participation in a project for the Dutch Ministry of Defence there was a sudden order resulting in all external staff, through financial difficulties, having to leave the (sizeable) project at the end of the month in question. The author was in the middle of his activities and therefore not able to conclude his part in the project properly.

In the part of IT auditor, the author audited a few "troubled projects". Four audits will come up for discussion later on in this thesis. They are part of the portfolio.

1.4 PROBLEM DETAILS

Basic assumptions/limitations on the problem definition

- in this study, the literature as mentioned in paragraph 1.3.1 is used. No other publications will come up for discussion;
- the projects and audits from the portfolio of projects as stated in paragraph 1.3.2 are used in this study. In addition, the "Netherlands Court of Audit" case is researched. No other projects are researched;

This indicates how the problem is defined.

1.5 APPROACH TO AND REPORT OF THE STUDY

The study is aimed at management of the projects in the portfolio of projects. This is achieved using success and failure factors that are derived from the literature as mentioned in paragraph 1.3.1. First, research is carried out for the benefit of the sub questions. Next, an answer to the definition of the problem is formulated.

Sub questions

1. *What is understood by success/failure factors in ICT (for short: SUFFIs = SUccess and Failure Factors in ICT projects)?*
   By means of the literature, it is investigated which definitions/descriptions are used by the researchers in question. Based on this, a selection is made which is used in this thesis.

2. *Is it possible to derive SUFFIs from international publications, Dutch publications and from the procedures in Tarek Abdel-Hamid's work on Software Project Management and if so, what particular SUFFIs?*
   The international and Dutch publications are studied and for each publication, the SUFFIs are represented in a table. The book by Tarek Abdel-Hamid & Stuart E. Madnick is studied. Each chapter is summarised in a number of characterising sentences. Based on these sentences, SUFFIs are determined and a SUFFI table is drawn up.

3. *Are there any Big Hitters amongst the SUFFIs and if so what are these?*
   An overview is drawn up of the Big Hitters as proposed by the researchers. Next, it is established whether these are indeed Big Hitters.

4. *Which SUFFIs are applicable to what particular project from the portfolio of the author's projects?*
   For every project from the portfolio, the applicable Big Hitters and a number of other SUFFIs are stated.

5. *Is it possible to use the Big Hitters for distinguishing between successful and not successful projects from the portfolio of the author's projects?*
   The projects and the matching Big Hitters are placed in an overview table. After analysis, this question is answered.

6. *Is it possible to present the SUFFIs in a SUFFI Chart that is easy to use in practice by others?*
   It is investigated whether the SUFFIs from sub question 4 can be included in one single table, in the form of a sort of checklist that can be used in practice.

7. *Is it possible to apply the SUFFI Chart in the "Netherlands Court of Audit" case?*
   The "Netherlands Court of Audit" report is studied. Each chapter is summarised in a number of characterising sentences. Based on these sentences, SUFFIs from the SUFFI Chart are linked. The same happens with the comments as provided by the various different experts. Next, conclusions are drawn.

Contents of the various chapters

The sub questions 1 up to and including 7 come up for discussion in chapters 2 until 8. Chapter 9 summarises the experiences of the author and chapter 10 contains the conclusions and recommendations. The addendums consist of a three appendices and 14 journal publications, including two articles that were published within the framework of this study.

Table 1.1: Portfolio of projects

No	Project with external project-based publications	Company / Remarks	Period	Duration In years	Ref. number In appendix external publications
1	POTVIS	National Police Services Agency (KLPD) • *Improvement to infrastructure in a complete SAP-environment: network segregation, RBAC, backup and restore, etc. (109 improvements).* • *Presentation of the results of RBAC at KLPD for the Dutch Society for Information Science NGI (80 participants, sheets are available).*	2003-2004	1.7	3
2	Kolibrie	KPN (Dutch Telecom company) • *Interfacing BAAN-ERP with other information systems (based on Enterprise Application Integration: message broker, adapters).* • *Presentation of the results of EAI at KPN for the Dutch Society for Information Architects (90 participants, sheets are available).*	1999-2001	2	4 and 5
3	Charging method GAK	GAK/ASZ • *Development of a charging method based on functional services (e.g. functional transaction codes).* • *Developed for the use with National databases (containing employees, employers and their contractual obligations).*	1997-1998	0.5	6
4	IMPALA	Delft University of Technology • *Implementation of a new PABX with 7000 telephone connections.* • *Charging modules, management procedures, information management, etc.* • *Project-based thesis Executive Master of IT Auditing (EMITA) EUR.*	1993-1994	2	7
5	OKAPI	University of Amsterdam • *Development of a new decentralised information system concerned with the financial building administration.* • *Including two-phase RBAC.*	1991-1994	2	8
6	GIRAF	Delft University of Technology • *General Information Retrieval Facilities on moinframe computers.* • *Many applications on the DUT.* • *Flexible multi-language informotion system.* • *Adaptations possible without changing the system.* • *Sold to other Universities and companies in the Netherlands ond Belgium.* • *After the project-based publication, some companies in the Netherlands & Belgium adapted the architectural/engineering solutions for all applications in their company.*	1982-1984	2.5	9 and 10
7	AUBID	Delft University of Technology • *Information system related to book reservations and book requests.* • *Based on CICS/VS.*	1977-1980	2.5	11 and 12

		• *Some solutions, published in the project-based publication, are implemented by dozens companies (especially the solution of the printer problem).*			
8	VDV	Delft University of Technology • *Unique system to collect traffic data on behalf of the Research of Traffic Streams.* • *Very difficult to develop (hardware, software).* • *Engineering system for the traffic engineers of the DUT.* • *Has been used for more than 25 years.*	1972-1974	3	13
9	BIBLIOSYSTEM	Delft University of Technology • *Batch oriented information retrieval system.* • *Many applications, including bibliographies.* • *After the project-based publication, more than 100 companies asked for the documentation/sources.*	1970-1971	0.5	14
	Project-based Audits				
10	Audit Multihouse	• In the nineties, Multihouse Automatisering B.V. developed a substantial information system by order of the SamenWerkingsVerband NUMIS-2000. • Project NUMIS-2000. • Court of Justice in Amsterdam. • Commission Third Party Experts: Professor M. Looijen, Professor G. Nielen, ir. Aart van Dijk.	1997		
11	Audit SYSA (GOVERN)	• Within the ABC Service of public body GOVERN, Unit A (UA) and Unit B (UB) function amongst others • In August 2001, the report "SAP R/3 for replacement of PROGA and PROGB" was issued. The reasons were that the applications were obsolete and did not work integrated. • Stopping with SAP/SYSA? • Auditor: Aart van Dijk	2004		
12	Audit ACCINT (PUBLIC)	• Within public body PUBLIC there is amongst other things, work in progress on the ACCINT project. • Carry out an audit/investigation into ACCINT with the objective to provide an answer to the question "What went wrong?". • Auditor: Aart van Dijk	2004		
13	Audit SOX (FINANCE)	• As part of SOX, FINANCE a well-known financial institution in the Netherlands, carried out audits/tests on the IT infrastructure in the period mid August until mid December 2006. • Project audits. • Auditor: Aart van Dijk	2006		

CHAPTER 2

WHAT IS UNDERSTOOD BY A PROJECT FAILURE?

2.1 DEFINITIONS

A few definitions are:

- the term "failure" refers to projects that are cancelled without completion due to cost or schedule overruns or that run later than planned by more than 25 percent [Jones 1996-1];
- a failure is defined as any software project with severe cost or schedule overruns, quality problems or that suffers outright cancellation [May 1998];
- Flowers [1996] defines an information system as a failure if any of these following situations occurs:
  1. When the system as a whole does not operate as expected and its overall performance is suboptimal.
  2. If, on implementation, it does not perform as originally intended or if it is so user-hostile that it is rejected by users underutilised.
  3. If the cost of the development exceeds any benefits the system may bring throughout its useful life.
  4. Due to problems with the complexity of the system, or the management of the project, the information system development is abandoned before it is completed;
- a successful project satisfies three factors: it complies with the functionality agreed to in advance, it is delivered on time and it is delivered within the agreed budget. When these three factors balance each other, we can speak of a successful project [Noordam et al. 2007];
- a definition of a failed project is quite easy – it is a project which does not make the journey from conception through to successful implementation [Smith 2001].

It may be concluded that the definitions are not univocal. Some authors are of the opinion that a project fails when it does not achieve successful implementation, other authors take this further and include the success of the information system in the organisation in relation with the user satisfaction and the benefits for the business in their assessment.

2.2 TAXONOMY OF PROBLEM PROJECTS

In his book, John Smith [2001] provides a "Taxonomy of Problem Projects" (figure 2.1).
In his book he is going to distinguish between projects which are *troubled* and projects which *fail*. A definition of a failed project is quite easy: "it is a project which does not make the journey from

conception through to successful implementation" [Smith 2001]. He gives an example.

> I was once given two weeks to prepare a set of 'scripts' for use at a dozen or more locations around the world. By following the scripts, unskilled staff would be able to migrate all user data from a variety of desktops and servers – many with different architectures and operating systems – to the 'strategic' replacement desktops and servers, which would have a common architecture. Users would be able to access all their data files on their new PCs and servers using the latest versions of applications. A salesman had observed that the Buyer's incumbent Vendor was approaching data migration in different ways at different locations and that skilled staff were being used. He made an assumption that our company could undertake the work more cheaply using unskilled staff and standard scripts. He sent the Buyer three presentation foils which represented an offer to migrate all their global data at a very low, fixed price per desktop. I looked into the assumptions which underpinned our proposal to the Buyer and found them to be flawed. Data migration could not be approached in this way. The project was basically not feasible and, if the offer was accepted, the project would represent a huge risk to our company. The proposal was withdrawn.

This is an example of a project which failed virtually at conception. Others might fail during design, development or implementation. Some projects travel right through their life cycle to operational use; it then becomes clear that the system delivered by the project does not, and never can, meet the business needs and is quickly withdrawn from service [Smith 2001].

John Smith: "The focus of my book is troubled projects and I shan't be saying anything more about failed projects, other than to point out that all failed projects were once troubled projects. It is our goal to detect and correct the causes of a project's troubles before it fails."

John Smith starts by discussing and categorising terms used by the leading authors in the field of troubled projects:

- John Boddie, in his book *Crunch Mode* [1987], describes projects which have very challenging schedules. They are achievable, just, but only through the use of very skilled people, using an approach which breaks some of the traditional 'rules' of system development;
- Ed Yourdon, in his book *Death March* [2004], considers projects in rather worse shape than this. A death march project is one which 'project parameters' (schedule, staff, budget and functionality) exceed the norm by at least 50 percent. Most 'death march' projects end in failure;
- 'Runaway Project' is a term coined by KPMG Management Consulting UK. KPMG defines a runaway as a project which has significantly failed to meet its objectives and/or is more than 30 percent over budget. The implication is that some 'runaways' will eventually be successful;
- Robert Glass uses the same term in his book *Software Runaways* [1998], although Glass uses a slightly different definition. He believes that a project is a true 'runaway' only if it is over budget by at least 100 percent.

John Smith: "If we were to consider a continuum of 'predicted cost to complete' divided by 'original budgeted cost' it might look a little like figure 2.1".

The meaning John Smith extract from figure 2.1 is that:

- "most projects get into crunch mode at some stage when the chips are down, but most projects with the potential to succeed eventually do complete;
- 'death march' and 'runaway' projects are hard to separate, but most 'death march' projects fail;
- troubled projects include some 'crunch mode' projects and all 'death march' and 'runaway' projects. I know one project very well wbich is five times over budget. I believe that it will complete successfully. It is certainly a 'runaway' and it has certainly been a 'death march' for the team, many of whom have been on the project for five years."

The next paragraph will further discuss troubled projects and failure factors in troubled projects according to John Smith.

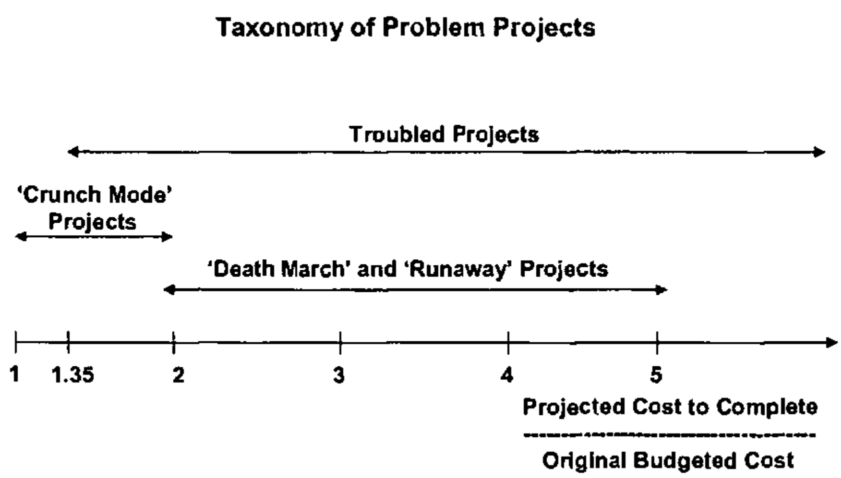


Figure 2.1: A Taxonomy of Problem Projects [John Smith 2001]

2.3 TROUBLED/FAILED PROJECTS

2.3.1 Definition of a troubled project

It is necessary to define a troubled project in more than simply qualitative terms, but a problem with quantitative measures is that they don't always scale. For example, if a three-month project takes six months, it may not be a total disaster for either the vendor or the buyer – provided that no major time imperative has been breached and the system is successful in operation. If a four-year project takes eight years to complete, this implies expenditure of a totally different magnitude.

John Smith's best shot at a definition of a troubled project is shown in the panel:

Definition

> A *troubled project* has one or more of the following characteristics:
>
> 1. it exceeds the planned time-scale by more than 50%, excluding the time-scale impact of agreed changes in scope;
> 2. it exceeds the build cost by more than 35%, excluding the cost of agreed changes in scope;
> 3. it is the cause of major buyer dissatisfaction to the extent that the future of the project is called into question;
> 4. the buyer lacks the commitment to make the project succeed;
> 5. it substantially fails to support the intended business processes;
> 6. it substantially fails to deliver the anticipated benefits;
> 7. the outcome for buyer-vendor is not win-win.

Example

> John Smith: "I once managed a project to design and develop a criminal information system for a UK police force. We had underestimated the amount of main memory required in the server. The Buyer did not see why they should pay for additional memory – after all, this was a fixed-price project. To save money, I took a decision to code the application in assembler rather than a high-level language so that the code would fit the available memory. This saved me money in the short term, but both timescale and budget were greatly exceeded as a result of this decision. Also, the eventual system, although it performed superbly and was very robust, required very specialised skills to maintain. This project exemplifies troubled project characteristics 1, 2 and 7."

2.3.2 Definition of a project failure

A successful project satisfies three factors: it complies with the functionality agreed to in advance, it is delivered on time and it is delivered within the agreed budget. When these three factors balance each other, we can speak of a successful project [Noordam et al. 2007].

For this thesis a project failure has one or more of the following characteristics:

1. it does not comply with the functionality agreed to in advance, including agreed changes of scope;
2. it exceeds the planned time-scale by more than 50%, excluding the time-scale impact of agreed changes in scope;
3. it exceeds the build cost by more than 50%, excluding the cost of agreed changes in scope.

2.4 DEFINITION OF THE PROBLEM [Answer to sub question 1]

Definition of the problem, sub question 1:

What is understood by success/failure factors in ICT (for short: SUFFIs = SUccess and Failure Factors in ICT projects)?

Answer:

In this paragraph the definition of a project failure is given. A success factor in ICT is a factor that contributes to the successful realisation of a project, for example “Good project management” or “Excellent definition of requirements”. A failure factor in ICT is a factor that contributes to a “project failure”, for example “Poor project management” or “Bad definition of requirements”.

CHAPTER 3

WHAT DO OTHERS THINK OF ICT PROJECTS FAILING OR NOT FAILING?

3.1 INTRODUCTION

In order to be able to establish how the projects the author (AvD) worked on agree or disagree with what others say happens, it is necessary to map out a number of publications (what others say happens) in this field. To that purpose, the following publications will be discussed:

International publications

- Large Software System Failures and Successes [Jones 1996-1]
- Major Causes of Software Project Failures [May 1998]
- Critical Success Factors In Software Projects [Reel 1999]
- Seven Characteristics of Dysfunctional Software Projects [Evans et al. 2002]
- The 40 root causes of troubled IT projects [Smith 2002]
- Critical failure factors in information system projects [Yeo 2002]

Dutch publications

- Why are information systems still failing? [Ooniricx 1982]
- Success and failure factors in complex ICT projects [Beenker 2004]
- ICT project management on the road to adulthood:
  Success factors for ICT projects [Noordam et al. 2007]

Tarek Abdel-Hamid

The procedures of Tarek Abdel-Hamid and Stuart Madnick in:
"Software Projects Dynamics – An Integrated Approach" [Abdel-Hamid & Madnick 1991].

The authors of these publications report on research carried out in the field of success and failure factors of ICT projects. Some authors have also published books on this subject (for example Abdel-Hamid, Smith, Ooniricx, Jones). For that reason, this collection of publications may also be considered representative for this subject as far as this thesis is concerned.
In section 3.2, "The 40 root causes of troubled IT projects" by John Smith [Smith 2002] come up for discussion. Next, sections 3.3 up to and including 3.6 discuss the following perspectives: "the size of

projects" [Jones 1996-1], "project risks based" [Evans et al. 2002], "other authors" [Yeo 2002, May 1998, Reel 1999] and "the Dutch situation" [Ooonincx 1982, Beenker 2004, Noordam et al. 2007]. This is followed by a summary in section 3.7. In section 3.8, the procedures of Tarek Abdel-Hamid and Stuart Madnick [Abdel-Hamid & Madnick 1991] are extensively discussed and the SUFFI model is presented.

3.2 THE PROJECT LIFE-CYCLE AND ROOT CAUSES OF TROUBLED PROJECTS

3.2.1 Introduction

Some ICT projects are doomed before the ink is dry on the contract. Others fall prey to troubled project root causes which strike later in the project life-cycle. Every project passes through a number of stages, as shown by John Smith in figure 3.2.1. The figure shows that most projects move through six stages. These stages often overlap to some extent. The stages are:

- Project Conception;
- Project Initiation/Mobilisation;
- System Design;
- System Development;
- System Implementation;
- System Operation, Benefit Delivery, Stewardship and Disposal.

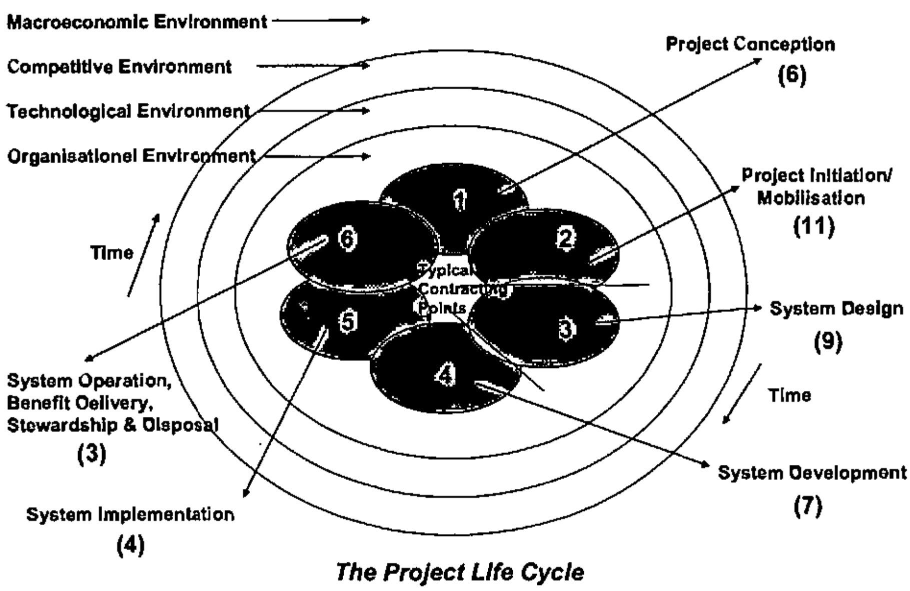


Figure 3.2.1: The Project Life Cycle [source: John Smith]

ICT engagements, projects and programmes bave a babit of getting into difficulty. If the problems are acute, the project will fail and both buyer and vendor may suffer substantial loss. More likely, it will become 'troubled', usually implying that it completes, but costs greatly exceed those budgeted and project time-scale is substantially longer than planned.

John Smith: "The 'soft' and 'business' skills needed to successfully engage with clients and safely contract for services are seldom taught on graduate courses; these skills must be acquired 'on the job'. This takes time and organisations have limited success at passing on 'lessons learned' ".

It is an interesting fact that, after analysing several sources of information and his own experience, John Smith was unable to find more than 40 generic root causes of troubled projects.

His list is shown in table 3.2.1.

Root causes (RC) of troubled projects

Author	No	Description
Project Conception		
JS	RC01	Project based on an unsound premise or an unrealistic business case
JS	RC02	Buyer failure to define clear project objectives, anticipated benefits and success criteria
JS	RC03	Project based on state-of-the-art and immature technology
JS	RC04	Lack of buyer board-level ownership/commitment or competence
JS	RC05	Buyer's funding and/or time-scale expectations unrealistically low
JS	RC06	Buyer failure to break a complex project into phases or smaller projects
Project Initiation/Mobilisation		
JS	RC07	Vendor setting unrealistic expectations on cost, time-scale or vendor capability
JS	RC08	Buyer failure to define and document requirements (functional and non-functional)
JS	RC09	Failure to achieve an open, robust and equitable buyer-vendor relationship
JS	RC10	Vendor failure to invest enough resources to scope the project to contract
JS	RC11	Buyer lack of sufficient involvement of eventual end-users
JS	RC12	Vendor underestimation of resources (predominantly person-effort) required
JS	RC13	Vendor failure to define project tasks, deliverables and acceptance processes
JS	RC14	Failure to actively manage risks and maintain robust contingency plans
JS	RC15	Poor project planning, management and execution
JS	RC16	Failure to clearly define roles and responsibilities in the contract/subcontracts
JS	RC17	Full-scope, fixed-price contracting (requirements, design and development)
System Design		
JS	RC18	Failure to 'freeze' the requirements baseline and apply change control
JS	RC19	Poor choice of technical platform and/or architecture
JS	RC20	Vendor starting a phase prior to completing a previous phase
JS	RC21	Poor choice of design/development method
JS	RC22	Failure to undertake effective project reviews and take decisive action
JS	RC23	Vendor lack/loss of skilled resources
JS	RC24	Poor vendor standards deployment (design, coding, testing, configuration management, etc.)
JS	RC25	Poor vendor requirements traceability (requirements > design > code > test)
JS	RC26	Buyer retention of design authority with right to approve/reject low-level designs
System Development		
JS	RC27	Delays causes the project to be overtaken by advances in technology
JS	RC28	Vendor failure to 'freeze' the design (and technical platform) and apply change control
JS	RC29	Inadequate vendor training and supervision of junior staff

JS	RC30	Inadequate vendor review of designs/code/documentation
JS	RC31	Poor vendor management of sub-contractors
JS	RC32	Lack of a formal, 'engineering' approach to integration and testing by vendor
JS	RC33	Insufficient attention paid by veodor to non-functional requirements
System Implementation		
JS	RC34	Buyer failure to manage the change implicit in the project (people, processes, technology)
JS	RC35	Inadequate user/systems training
JS	RC36	Catastrophic failure of the system, with no effective contingency arrangement
JS	RC37	Missing a crucial 'go live' date
System Operation, Benefit Delivery, Stewardship and Disposal		
JS	RC38	Buyer failure to measure actual delivered benefit and take corrective action
JS	RC39	Buyer failure to maintain/enhance system post-implementation
JS	RC40	Changes in the competitive or macro-economic eovironment

Table 3.2.1: Root causes (RC) of troubled projects

Which root causes are the 'big hitters'?

Any of the root causes can have a profound impact on the outcome of a project, and the problem is that they hunt in packs. Many troubled projects are beset by, perhaps, half a dozen of these root causes. It is unwise to try to rank the root causes into 'big hitters' and 'the rest'. However, such ranking is definitely of value for an individual project as it will inform the prioritisation of 'turnaround' actions. With regard to this particular point, John Smith disagrees with other researchers [Evans, Yeo, Reel, Beenker, Noordam] that do indicate a ranking of root causes.

3.2.2 Government ICT projects

Whilst it is true to say that government ICT projects are procured and managed rather differently from private sector projects, the report "Improving the delivery of government ICT projects" [House of Commons 1999] has much valuable advice to offer. A number of "war story projects" are exposed. Summarising the key conclusion and recommendations of the report finds the following root causes of troubled projects in the UK public sector:

Author	No	Description
JS	PUBRC01	Lack of senior management involvement and commitment
JS	PUBRC02	Failure to focus on key business and end-user needs
JS	PUBRC03	Failure to break complex projects into manageable, separately contracted 'chunks'
JS	PUBRC04	Poor and unimaginative project management
JS	PUBRC05	Poor risk management and contingency planning
JS	PUBRC06	Unclear contracts and poor contract management
JS	PUBRC07	Insufficieot focus on user training needs and the design of training interventions

Table 3.2.2: Root causes (RC) of troubled projects in the UK public sector

3.2.3 Benefits

There are three benefits to be derived from consideration of troubled project root causes:

- better solutions and proposals;
- early warning of problems in delivery;
- the potential to turn around a troubled project.

3.3 THE SIZE OF PROJECTS

3.3.1 Introduction

An interesting question is whether the size of a project does play a part in success and failure factors, because a large project may result in a higher degree of complexity and a longer turn around time.

Failure factors mentioned by John Smith are for example: "Buyer failure to break a complex project into phases or smaller projects" (JS/RC06) en "Failure to break complex projects into manageable, separately contracted 'chunks' " (JS/PUBRC03).

Professor Nielen [Nielen & Nielen 1996, Van Dijk et al. 1997] has the opinion that the realisation-effort of a project relates quadratically to the size of the project. An information system of for example 6,000 function points is 9 times more difficult to realise than an information system of 2,000 function points.

Peter Noordam et al. [2007]: "We have looked at another important success factor: the project size. Looking at the project size in our study we see significant differences between banks and insurers on the one hand and the central government on the other hand. Very large projects worth more than 10 million Euros only take place within the central government; 64% of all projects at banks and insurers are worth less than 1.5 million Euros, whilst this is about 43% for the central government projects. This could explain why the central government is less satisfied as compared to banks and insurers. From the research carried out by the Standish Group International Inc [1998] it can be concluded that it is sensible to set up projects such that these do not exceed thc 1.5 million dollar line in order to increase the chance of success (see table 3.6.4.1)."

In 1996, Capers Jones carried out research into "Large Software System Failures and Successes". In this research, he did not just look at the size of a project, but also studied the type of industry for which the software was developed. The results are shown below.

Capers Jones: "Computers and software have become indispensable in modern business, government, and military operations. The need for software has created one of the major occupations of the 20th century. However, software is a troubling technology. Software development is highly labour intensive, and as a result, large software projects are among the most expensive undertakings of the 20th century. For example, large software systems cost far more to build and take much longer to construct than the office buildings occupied by the companies that have commissioned the software".

3.3.2 Size of project

Really large software systems in the 100,000 function point (FP) size range can cost more than building a domed football stadium, a 50-story skyscraper, or a 70,000-ton cruise ship. Not only are large systems expensive, but they also have one of the highest failures rates of any manufactured object in human history. The term "failure" refers to projects that are cancelled without completion due to cost or schedule overruns or that run later than planned by more than 25 percent. Capers Jones: "Let us consider what the phrase 'large systems' means in the context of six different size plateaus separated by an order of magnitude for each plateau:

- 1 Function Point (125 C statements);
- 10 Function Points (1,250 C statements);
- 100 Function Points (12,500 C statements);
- 1,000 Function Points (125,000 C statements);
- 10,000 Function Points (1,250,000 C statements);
- 100,000 Function Points (12,500,000 C statements)."

1 Function Point

There are few software applications of this size, except small enhancements to larger applications or minor personal applications. The schedules for such small programs usually range from only a day to perhaps a week. In the Capers Jones study, 1 Function Point is put on a par with 125 statements from computer language C.

10 Function Points

This is the typical size of end-user applications and also a very frequent size plateau for enhancements to existing software. Development schedules are usually less than a month.

100 Function Points

This is the practical upper limit of end-user applications. Development schedules are usually less than six months. Individual programmers can handle applications of this size, although technical writers and other specialists may be involved too.

1,000 Function Points

This size range exceeds the capabilities of end-user development. Schedules for software projects of this size are usually longer than 12 months. Quality control is a major requirement at this size range. Applications in this size range require development teams of up to 10 staff members. With team development, issues of system segmentation and interfaces among components become problematic.

10,000 Function Points

Applications of this size are usually termed "systems", because they are far too large for individual programs. This size range is often plagued by cost and schedule overruns and by outright cancellations. Development teams of 100 or so are common, so communication and interface problems are endemic. Software schedules in this size plateau run from three to more than five years, although the initial planning for applications of this size range tends naively to assume schedules of 18 months or less. Configuration control and change management are also mandatory for this size plateau.

100,000 Function Points

Applications that approach 100,000 function points in size are among the most troubling constructs of the 20th century. This is roughly the size range of Microsoft's Windows 95 product and IBM's MVS operating system. This is also the size range of major military systems. Software development schedules for systems of this size usually range from five to more than eight years, although the initial development plans tend to assume 36 months or less. Communication problems are rampant. Formal configuration control and change management are mandatory and expensive for this size plateau.

Capers Jones: "Using these six size ranges, table 3.3.1 shows the approximate frequency of various kinds of outcomes, ranging from finishing early to total cancellation [1996-1]."

	Probability of Selected Outcomes				
	Early	**On time**	**Delayed**	**Cancelled**	**Sum**
1 FP	14.68%	83.16%	1.92%	0.25%	100.00%
10 FP	11.08%	81.25%	5.67%	2.00%	100.00%
100 FP	6.06%	74.77%	11.83%	7.33%	100.00%
1.000 FP	1.24%	60.76%	17.67%	20.33%	100.00%
10.000 FP	0.14%	28.03%	23.83%	48.00%	100.00%
100.000 FP	0.00%	13.67%	21.33%	65.00%	100.00%
Average	5.53%	56.94%	13.71%	23.82%	100.00%

Table 3.3.1: Software project outcomes by size of project

3.3.3 Six sub industries

Capers Jones: "Size is not the only factor that influences the outcome of software projects. A surprisingly strong influence is the nature of the industry that builds the application. Some industries are much better than others in controlling large software systems, although no industry is perfect. There are six major sub industries within the software community that tend to follow somewhat different practices and even use different tools and programming languages. In terms of their ability to build large software applications successfully, the rank order of these six sub industries is:

1. Systems software;
2. Outsource vendors;
3. Commercial software;
4. Military software;
5. Management information software;
6. End-user software".

Table 3.3.2 sbows the probabilities of on-time development for the six software domains.

	Systems software	Outsrc. software	Comm. software	Military software	MIS software	End-user software	Average
1 FP	99.00%	98.00%	99.00%	98.00%	98.00%	95.00%	97.83%
10 FP	96.00%	97.00%	98.00%	93.00%	95.00%	75.00%	92.33%
100 FP	88.00%	88.00%	89.00%	84.00%	86.00%	50.00%	80.83%
1,000 FP	75.00%	74.00%	75.00%	65.00%	68.00%	5.00%	60.33%
10,000 FP	54.00%	47.00%	35.00%	38.00%	30.00%	---	40.80%
100,000 FP	28.00%	24.00%	10.00%	15.00%	5.00%	---	16.40%
Average	73.33%	71.33%	67.67%	65.50%	63.67%	56.25%	66.88%

Table 3.3.2: Probability of on-time software delivery in six sub industries

Note: There are no end-user applications larger than 1,000 function points, so the "---" values in the end-user column are excluded from the average values.

3.3.4 Conclusions

Attempting to construct large software projects without adequate management and quality control is not a safe undertaking. No one in the industrialised world today would dream of starting a large engineering project without adequate tools for project management. Software itself is intangible, but the schedules and cost estimates for software can be highly tangible. Software projects are still subject to the basic laws of manufacturing and software needs to be placed on a firm engineering basis by the end of the 20th century.

3.4 PROJECT RISKS BASED

3.4.1 Introduction

Taking advantage of its many years of experience in identifying and evaluating project risks in large-scale software systems acquisition and development programs, Integrated Computer Engineering (ICE) has developed a risk database. Their analysis of the risk database has identified seven predominant characteristics that provide insight into the causes of dysfunctional software projects. This paragraph identifies these characteristics and the typical real-world risks that accompany each [Evans et al. 2002]. Michael Evans: "The project risks that were collected during ICE's 12 years of project assessments began to form a substantial database of useful information. Also added to this risk database are project risks gathered from risk studies conducted by Institute for Defence Analysis [1994], risks identified by Capers Jones [1994] and Tom DeMarco [1995]. To date, the ICE risk database has grown to more than 800 primary and secondary project indicators [Software Program Managers Network 2000]. What emerged from the ICE risk database were seven predominant characteristics relating directly or indirectly to common failures observed among those system acquisition and development projects that had the greatest difficulty delivering a quality product on time and on budget."

In paragraph 3.4.2, the seven predominant characteristics are listed. Some real-world risks that have been associated with the seven characteristics are described in 3.4.3.
The links between the characteristics and the real-world risks are made in paragraph 3.4.4.

3.4.2 Characteristics

The seven common characteristics are [Evans et al. 2002]:

1. Failure to Apply Essential Project Management Practices;
2. Unwarranted Optimism and Unrealistic Management Expectations;
3. Failure to Implement Effective Software Processes;
4. Premature Declarations of Victory;
5. Lack of Program Management Leadership;
6. Untimely Decision-Making;
7. Lack of Proactive Risk Management.

3.4.3 Risk designators

The 18 risk designators are:

- the process being followed and decisions being made will result in a product that may not satisfy the critical needs of the user and are inconsistent with the severity of the consequences of project failure (A);

- the staff are not capable of implementing the product and applying the technologies selected. Excessive turnover may impact project success (B);
- project plans are unrealistic or not implemented and do not result in a predictable development environment (C);
- design and code defects will not be discovered until late in the development – too late to avoid cost, schedule or quality impacts (D);
- the plan has not been updated recently and now is out of date with the project environment and does not reflect current agreements or constraints (E);
- there is inefficient software development due to failure to allocate requirements early in the design phase (F);
- the technical process being used is inconsistent with the project's requirements and the staff's ability to implement it (G);
- the design may not support the application's critical safety or security requirements (H);
- project plans do not describe how technology will be used resulting in a need to continuously rework inconsistent products and correct resulting problems (I);
- software reliability problems will not be discovered because procedures are not established for the collection and analysis of error data generated during software development (J);
- software defects will not be found because the contractor has neither conducted nor planned for software design inspections or walk throughs (K);
- essential systems functions do not perform adequately or reliably due to testing problems or insufficient testing of key software components (L);
- customer relationships cause an environment that is unstructured and precludes successful implementation of a product within cost and schedule (M);
- the system may not satisfy the needs or expectations of the user when delivered (N);
- early release of unqualified products results in unexpected failures, failures in key user areas, and potentially corrupted data, which destroys confidence in future releases (O);
- miscellaneous risks not being tracked make project success unlikely (P);
- risk management may not prove effective or identify key risks (Q);
- the lack of an effective risk management process results in unplanned problems impacting the project (R).

3.4.4 Characteristics and their risk designators

Introduction

This paragraph (and table 3.4.1) shows each of the seven characteristics and their respective risk designators (an upper case letter over the range A to R).

Failure to Apply Essential Project Management Practices

Typical risk designators are: A,I,J,C,K,L

Michael Evans: "What we repeatedly find through assessments is that while the mainstream software tasks have been reasonably well planned and implemented, certain essential project management practices are not. Some managers perceive these practices as bureaucratic red tape that only gets in the way of real engineering."

Unwarranted Optimism and Unrealistic Management Expectations

Typical risk designators are: A,B,C

Michael Evans: "In some projects there is an underlying belief that all will be well.
In 1995, only 16 percent of software projects were expected to finish on time and on budget. An estimated 53 percent of projects cost nearly 190 percent of their original estimates."

Failure to Implement Effective Software Processes

Typical risk designators are: G,D,H,F

Michael Evans: "Many software projects' managers assume that since trained software engineers staff the project, project-specific standards, guidelines and common tools are unnecessary. The key to success is adaptation of the technology and process to meet the unique challenges of a specific project or program."

Premature Declarations of Victory

Typical risk designators are: L,N,O

A clear understanding of the customer's quality expectations is an essential prerequisite to client satisfaction. "The decision to pressure people into delivering a product that doesn't measure up to their own quality standards is almost always a mistake" [Alan Davis 1993].

Lack of Program Management Leadership

Typical risk designators are: C,E,M,B

Michael Evans: "Poor project management will defeat good engineering and is the most frequent cause of project failure. Attributes of a good software project manager include a broad range of technical software development experience, the ability to manage people and the dynamics of a team environment, and the willingness to proactively manage project risk and make timely decisions. To paraphrase Tom DeMarco [1995]: 'Managers ... make craziness go away' ".

Untimely Decision-Making

Typical risk designators are: D,E,F

Michael Evans: "Management is the art of planning work so that it can be accomplished within constraints of time, cost and other resources at a level that will be competitive in the marketplace".

Lack of Proactive Risk Management

Typical risk designators are: P,Q,R

Michael Evans: "The problem of project management, like that of most management is to find an acceptable balance among time, cost and performance".
"Projects that fail to do an effective job of managing risk are constantly reacting to problems, while those that manage risk well anticipate rather than react. 'Your organisation will be much better once it moves away from reacting to change, and toward proactive anticipation and management of change' ".

3.4.5 Analysis and conclusions

Table 3.4.1 shows each of the seven characteristics and their respective risk designators (an upper case letter over the range A to R). Each risk event in the ECI database was characterised against the risk designators, and the tallied results are shown in the third column from the left. The risk designator events were accumulated for each risk characteristic (fourth column) and the frequency of occurrence relative to all observed events in the database was calculated (far right column). It should be noted that the percentages in the far right column do not total 100 percent, as the risk designators are not unique to each characteristic. The ranking of the characteristics is by frequency of characteristic occurrence; therefore, *the data show what may be the likely dysfunctional causes, but not their relative impact on projects or programs.*

Michael Evans: "When reviewing dysfunctional software projects, a reasonable approach would he to consider the risk descriptions for each of the seven characteristics we have identified and determine whether they apply. Why do projects not address these issues if they are so apparent? The first reason is denial. Denial is the excuse that enables program managers to make dumb decisions. The second reason concerns cultural barriers. *Coincidentally, all of the seven factors we identified focus on cultural, rather than technical issues*. 'For the overwhelming majority of the bankrupt projects we studied, there was not a single technological issue to explain the failure'. Factors such as the seven addressed here do matter and they should he considered essential components of any project."

Characteristic	Risk Designator	Number of risk events applicable to specific Risk Designator	Number of risk events for Characteristic	Frequency of occurrence relative to all observed risk events (see note 1)
1. Failure to Apply Essential Project Management Practices	A	246	480	57%
	I	6		
	J	36		
	C	66		
	K	10		
	L	116		
2. Unwarranted Optimism and Unrealistic Executive Management Expectations	A	246	344	41%
	B	32		
	C	66		
3. Failure to Impl. Effective Software Processes	G	162	248	30%
	D	15		
	H	26		
	F	45		
4. Premature Declarations of Victory	L	116	165	20%
	N	46		
	O	3		
5. Lack of Program Management Leadership	C	66	106	13%
	E	3		
	M	5		
	B	32		
6. Untimely Decision-making	D	15	63	8%
	E	3		
	F	45		
7. Lack of Proactive Risk Management	P	4	24	3%
	Q	9		
	R	11		

Note 1: The total number of risk events categorised (841 events) was used as the baseline population of risk events for frequency of occurrence calculations

Table 3.4.1: Seven Characteristics of Dysfunctional Software Projects [source: Michael Evans et al. 2002]

3.5 OTHER AUTHORS

3.5.1 Introduction

This paragrapb gives the success and failure factors of some authors without extensive foundation. Together with the success and failure factors as already mentioned and the ones in the next paragrapb, these will be used in paragrapb 3.8.6 for arriving at a *success and failure factars model* that can be used in cbapter 5 when discussing the portfolio projects and other practical observations.

3.5.2 Yeo (2002)

Yeo [2002] starts from a triple-S framework tbat consists of Process driven issues (Sp), Context driven issues (S1) and Content driven issues (S2). The triple-S framework can be perceived as three *spheres of influence* (SOI) over project outcomes. The three SOI are further made operational into 10 main issues of influence (IOI), as sbown in table 3.5.1. These issues are in turn defined by a list of failure factors filtered out from an in-depth literature review on information system and system failure studies.

No	
	Sp Process driven issues
1	Business planning
2	Project planning
3	Project management and control
	S1 Context driven issues
4	Corporate culture
5	Corporate management
6	Users
7	Politics
	S2 Content driven issues
8	Information technology
9	Business process and system design
10	IT/IS professional and knowledge sources

Table 3.5.1: Defining "Issues of influence" under the three "Spheres of influence"

Yeo: "Failure factors analysis is based on 92 usable returns from a Singapore-based questionnaire survey conducted in November 2000. Table 3.5.2 lists the top five factors categorised under Sp, S1 and S2 sphere of influence respectively" [Yeo 2002].

Rank	Sp Process driven issues	S1 Context driven issues	S2 Content driven issues
1	Underestimate of timeline	Lack user involvement and inputs from the onset	Consultant/vendor underestimated the project scope and complexity
2	Weak definitions of requirements and scope	Top down management style	Incomplete specifications when project started
3	Inadequate project risk analysis	Poor internal communication	Inappropriate choice of software
4	Incorrect assumptions regarding risk analysis	Absence of an influential champion and change agent	Changes in design specifications late the project
5	Ambiguous business needs and unclear vision	Reactive and not pro-active in dealing with problems	Involve high degree of customisation in application

Table 3.5.2: Top 5 failure factors under Sp, S1 and S2

3.5.3 May (1998)

Lorin May: "Most software projects can be considered at least partial failures because few projects meet all their cost, schedule, quality, or requirements objectives. Failures are rarely caused by mysterious causes, but these causes are usually discovered post-mortem, or only after it is too late to change direction". The results of Lorin May in this paragraph are based on interviews with software consultants and practitioners who were asked to provide 'autopsies' of failed projects with which they have been acquainted.

According to the Standish Group [1995] U.S. government and businesses spent approximately $ 81 billion on cancelled software projects, and another $ 59 billion for budget overruns. Their survey claimed that in the United States, only about one-sixth of all projects were completed on time and within budget, nearly one third of all projects were cancelled outright, and well over half were considered 'challenged'. Of the challenged or cancelled projects, the average project was 189 percent over budget, 222 percent behind schedule, and contained only 61 percent of the originally specified features [May 1998]. One reason for the varied conclusions is that most failed projects are never studied – even by the organisation that experienced the failure. Information about project failures often relies heavily on subjective assessments. Lorin May: "A failure is defined as any software project with severe cost or schedule overruns, quality problems or that suffers outright cancellation" [May 1998]. The failure causes are:

1. Poor user input;
2. Stakeholder conflicts;
3. Vague requirements;
4. Poor cost and schedule estimation;
5. Skills that do not match the job;

6. Hidden costs of going 'Lean and Mean';
7. Failure to plan;
8. Communication breakdowns;
9. Poor architecture;
10. Late failure warning signals [May 1998].

3.5.4 Reel (1999)

John Reel: "Throughout the fifty-odd years of software development, the industry has gone through at least four generations of programming languages and three major development paradigms. We have held countless seminars on how to develop software correctly, forced many courses into undergraduate degree programs, and introduced standards in our organisations that require specific technologies. Still, we have not improved our ability to successfully, consistently move from idea to product".
Software-based systems are exceptionally complex. In fact, many agree that 'the basic problem of computing is the mastery of complexity'. Add the fact that developers are trying to hit a moving target – user requirements – and you get a volatile mixture of management issues.
The Chaos study, published by the Standish Group [1998], also shows that the completion rate has improved because companies have trended towards smaller, more manageable projects – not because the management techniques have improved [Reel 1999].

Managing a successful software project

In software, more 'advanced' technologies are far less critical to improving practice than embracing what I believe are the five essential factors to managing a successful software project [Reel 1999]:

1. Start on the right foot;
2. Maintain momentum;
3. Track progress;
4. Make smart decisions;
5. Institutionalise post-mortem analyses.

If you master these five critical success factors, you greatly increase the odds of completing your project on time and within budget. Just as important, you increase your chances of actually delivering something your users want.

Start on the right foot

Just as it is difficult to grow strong plants in weak soil, it is almost impossible to successfully lead a development effort that is set up improperly. Tom Field analysed pitfalls in software development efforts and gave 10 signs of IS project failures:

1. Project managers don't understand users' needs;
2. The project's scope is ill-defined;
3. Project changes are managed poorly;
4. The chosen technology changes;
5. Business needs change;
6. Deadlines are unrealistic;
7. Users are resistant;
8. Sponsorship is lost;
9. The project lacks people with appropriate skills;
10. Managers ignore best practices and lessons learned [Reel 1999].

3.6 THE DUTCH SITUATION

3.6.1 Introduction

From a scientific point of view, the Netherlands has enjoyed an excellent reputation with respect to software research [Aerts 2006/2007] for many years. Occasionally, there were complaints about this research being very theoretically orientated. However, that is changing. The Ministries of Economic Affairs and Education Culture and Science and financial backer of research NWO decided in 2000 to give an impulse to software research with industrial relevance. Representatives from trade and industry and research institutes jointly set up a programme, which from 2002 was known as "JACQUARD" (Joint ACademic QUAlity Research and Development). JACQUARD resulted in an intensive interaction between researchers and a few dozen companies. By means of discussions on progress and user committees, the businesses ensure that research is embedded into everyday practice. An assessment committee consisting of renowned scientists and people from the business community assess the propositions for scientific quality, innovatively and practical relevance for the business community. Professor Chris Verhoef of the Free University Amsterdam: "The strategy of the Free University is that there is no science without social relevance. That also becomes evident from the way in which we view informatics" [Aerts 2006/2007]. It may be concluded that some universities/professors are clearly changing course from very theoretically based software research to software research that has industrial relevance. In the opinion of the author (AvD), introduction to the Netherlands of the Engineering Doctorate (EngD) after the British example, can substantially contribute to that.

In the Netherlands, research is/will be carried out into success and failure factors regarding ICT projects. As early as 1982, Professor Jan Oonincx wrote his book "Why are information systems still failing?". In his inaugural speech [Berghout 2002] at the Rijksuniversiteit Groningen in 2002, Professor Egon Berghout said: "This booklet is, without a doubt, based on many years of annoyance preceding 1982; however twenty years later it can be reprinted almost unchanged. Hardly any causes have been removed".

Recent studies in the Netherlands in this field are:

- Success and failure factors in complex ICT projects [Nico Beenker 2004];
- ICT project management on the road to adulthood:
  Success factors for ICT projects [Noordam et al. 2007].

Oonincx' book and the studies carried out by Beenker and Noordam et al. come up for discussion next.

3.6.2 The past

3.6.2.1 Introduction

In the Eighties, I wrote a large number of literature references for the PRISMA – Book review service in Voorburg. These references had to be brief and to the point. One of the very first books that I reviewed was "Waarom falen informatiesystemen nog steeds?" (Why are information systems still failing?) by Professor Jan Oonincx. That review, which I wrote 26 years ago, is shown below. In the Netherlands, Oonincx' book was, as far as I am aware of, the first book to determine factors determining success and/or failure. Many other writers also often refer to it (as seen in paragraph 3.6.1 and 3.6.3.1). At the time of writing that review, I had been working within the IT field for 17 years and when reading the book I often thought: "All this has been common knowledge for a long time!". The past 26 years have proved that many colleagues apparently still do not know all this or do not know how to act accordingly. So much becomes crystal clear, when reading this part of the thesis.

26 years ago …….. PRISMA – Book review service, Voorburg:

Although there are many well-functioning information systems, there are plenty of information systems that either function badly or never reach their original target as set during construction, namely being taken into service. This book discusses, grouped in seventeen chapters, a number of aspects that often cause development and introduction of information systems to fail. Information systems, which are set up too ambitiously, too isolated or without proper planning, stand a very large chance of failing. Insufficient involvement of future users in the development of information systems or a passive attitude of the top management also often leads to disappointing results. The book is suitable for students of information science, business administration, economics etc. Although the IT engineer, employed in practice will encounter a number of familiar cases, he will also benefit from reading the book attentively. (Future) users of information systems will largely reap the benefits from studying this book. In this case, the argument "no time" will not wash. It is a very readable book of modest proportions but with very important contents.
25 August 1982 - ir. Aart J. van Dijk

3.6.2.2 The subjects discussed by Oonincx

Professor Oonincx: "In this publication it is discussed why up to date (1982) so few successfully computerised information systems were realised [Oonincx 1982, Lucas 1975]. Through pointing out a number of responsible causes (the errors and shortcomings from the past), we do hope to contribute to different (adjusted) ways of approaching the development of information systems. In this book, the lack of a theory and methodology for developing information systems was selected as a starting point."
Table 3.6.2.1 contains an overview of "Failure factors in development and introduction of information systems". In appendix 1, the 17 chapters of Oonincx' publication will be discussed in brief.

Author	No	Description
JO	01	The lack of a methodology for developing information systems
JO	02	Insufficient insight into decision-making processes and information requirements
JO	03	The feasibility of integral information systems
JO	04	Insufficient involvement and cooperation of users in setting up information systems
JO	05	Insufficient and/or inefficient use of the possibilities as offered by the computer in the information system
JO	06	Not using the new information system
JO	07	Collection and storage of too much and unsuitable data
JO	08	Ineffectiveness and limited applications of traditional systems for data storage
JO	09	Set-up of information systems too static
JO	10	Insufficient suitable human resources for development and implementation of information systems
JO	11	The influence of changes to the information system on the organisation
JO	12	Too little attention to the informal aspects in relation to the information system
JO	13	The development process takes too long and the costs are too high
JO	14	Computerisation based on status considerations
JO	15	Too little attention to the aspect of profitability and the lack of measuring instruments for determining the benefits of the information system
JO	16	Unsuitable equipment and software
JO	17	Insufficient attention to reliability and controllability

Table 3.6.2.1: Failure factors in development and introduction of information systems [source: Oonincx 1982]

3.6.3 The opinion of the ICT lawyers

3.6.3.1 Introduction

As early as 1982, Jan Oonincx published a classic under the meaningful title: "Waarom falen informatiesystemen nog steeds?" (Why are information systems still failing?). Nico Beenker [2004]: "Over 20 years later (2004), this subject is as topical as it was then, because every single year sees many complex ICT projects failing". There have been a number of infamous cases in the Netherlands. In his own experience as a consultant, Nico Beenker sees complex projects both succeed as well as fail.

This often happens according to similar patterns. Nico Beenker: "The seeds for success and failure of complex ICT projects are often created early on in the process. Many ICT professionals will recognise that: you often have an idea how the project will progress at an early stage. It is often difficult, in spite of all theoretic insights, to get this under control. Because of my own observations, there is a growing conviction that other elements must play a part in this. In failing ICT projects, there is often more going on than just problems to do with project management, organisational changes and such like". His study consists of two parts, a literature search /desk research and field study in the form of a series of in-depth interviews. The following research questions take central stage:

- do indeed that many complex ICT projects fail in the Netherlands?;
- which factors play a part in the success and failure of complex ICT projects?

An important conclusion of this study is that the existing theoretical frameworks do not prove adequate solutions for many occurring problems that cause complex ICT projects to collapse. Nico Beenker discovered five aspects that are of significant influence to success and failure: "These aspects are, as far as I have been able to ascertain, up until now not been linked to ICT projects."

3.6.3.2 The problem of failing projects

Sometimes, it is said [Beenker 2004] that "*With regard to ICT projects you have to stick with the following simple rule of thumb: 2:2:1/2. They take twice as long as planned, are twice as expensive and you only get half of what you want*". ICT projects do indeed have a bad reputation, they are often too late, (too) expensive and sometimes one does not succeed in completing the project at all. The annual study into trends in ICT as carried out by Ernst & Young amongst 650 managing directors, managers and professionals [Verschuur 2003], does confirm this image, albeit slightly less seriously. According to the most recent version of the study, only about a third of all ICT projects in the Netherlands are delivered within the set time and budget.

How much goes wrong in the Netherlands?

The question how things are in the Netherlands presents itself. Beenker said: "It is not just interesting to find out how much actually goes wrong but also why ICT projects do fail. Customers and suppliers of the ICT industry do not like talking about failing ICT projects, all this is very embarrassing". An exception to this is the Foundation for the Settlement of Automation Disputes (Stichting Geschillen Oplossing Automatisering or SGOA). The objective of this foundation is mediation in, solving of and – if necessary – arbitrage in disputes in the field of ICT. The SGOA only handles about 20 disputes per year, representing a value of 7 million Euros [Vink 2002]. This corresponds with not even a tenth of a percent of the total investment volume in ICT. Going by SGOA's figures you could draw the conclusion that with regard to failing ICT projects things are not all that bad in the Netherlands. However, is that conclusion justified? If the percentages from the research done by The Standish Group, which in 2003 based itself upon the data of 13,522 American ICT projects, would apply to the Dutch situation, then 1.7 billion Euros in ICT projects would fail entirely every year (15%) and thousands of projects with a combined value of 5.9 billion Euros would have a dubious end result (51%). Of course, you cannot simply apply the figures of The Standish Group to the Dutch situation. However, these figures do fit in better with the more general coverage on the success and failure of complex ICT projects. Beenker: "The failing of ERP (Enterprise Resource Planning) and CRM (Customer Relationship Management) projects received particularly gloomy coverage in the last 10 years. In publications on the failing of ERP/CRM projects [De Vrede 2001, De Vrede 2002], estimations on failing projects range from 50% to 75%. Observations in my own experience as a consultant do confirm the last image" [Beenker 2004].

Complex ICT project [Beenker 2004]

In his study Beenker uses a broad definition of complex ICT projects: " Traditional system development takes central stage, in which the objective is (re)building and putting an ICT system into operation. Outsourcing projects have not been included in this research. An ICT project can be complex for several different reasons. Larger ICT projects are often complex simply because of their size and number of people involved. However, even small ICT projects can be complex, for example for technical or organisational reasons. This is the entire process in which a functioning (software) system is delivered including the required infrastructural, personnel and procedural or even organisational changes for enabling optimum exploitation of this system. With respect to this, it is important that this definition includes a role for the client. The reason being that it is the client who informs the provider on which processes need to be supported and which functional, technical and possible implementation requirements these need to meet. Which development approach is used in this has been ignored. It is relevant that an intensive cooperation between provider and client is a requirement at all times" [Beenker 2004].

3.6.3.3 The field study

Set-up

There is a group of professionals in the Netherlands, which has a unique perspective of failing ICT processes, namely the lawyers involved in IT law. I also appeared in court as an expert witness (see chapter 5). Nico Beenker [2004]: "We carried out field study amongst them, which consisted of a series of interviews on the subject 'success and failure factors in complex ICT projects'. The ground for specifically targeting this professional group results from the thought that where costly ICT projects fail or experience stormy weather, this will cause conflicts between parties and lawyers will be called in. At the start of 2004, we spoke with 16 specialised ICT lawyers. The interviews were semi structured, in other words conversations were started using only a rudimentary list of questions. The interview gradually gained more of an open character. On average, the sixteen interviews took about an hour and a half. Many sometimes sensational cases were passed in review. Of course, the names of the clients were not revealed, although it was sometimes possible to guess who they were".

Reliability and validity

The result of the field study is worthwhile because of the breadth and depth of the interviews. Nico Beenker: "About the reliability and validity of the field study, it can be said that the composition (just ICT lawyers) and the scope of the sample survey cannot be considered sufficiently reliable for answering the more quantitative research questions. Anyway, that was not the objective of the field study. Research bureaus like Gartner and Standish can do that much better."

Results of the field study

- one of the results of the field study was a long list of failure factors of ICT projects, however no new "practical theory" is to be expected from the legal corner;
- many more ICT projects fail in the Netherlands than was generally known up until now. The offices where the interviewed lawyers work, deal with hundreds of ICT cases every year, representing a total underlying value of over eight hundred million Euros (see table 3.6.3.1). In view of the above, the percentage figures of the "Chaos study" could certainly be applicable to the Netherlands. This brings the total of failing projects to 15% (1.7 billion Euros), whilst projects with a debatable end result make up almost half, representing a collective value of 5.9 billion Euros;
- only a very small part of these cases ends up in court. In most cases by far, things are settled amicably or a type of mediation takes place. Nico Beenker: "Shame and loss of reputation mean that parties tend to keep this matter behind closed doors."

Number of cases dealt with by the thirteen offices involved in this field study of a total project value of:	Average number per year dealt with*
Over 50,000 Euros	> 500
Over 3 million Euros	± 100
Over 100 million Euros	± 5
* estimation based on the last 3 years	

Table 3.6.3.1: Number of cases dealt with by the investigated lawyer's offices

- a number of failure factors do stand out because these are mentioned considerably more often than the other ones. Below a list of the most often mentioned factors; starting with the factor that is mentioned most often:

    1. planning is too optimistic;
    2. badly phrased contracts;
    3. poor project management;
    4. poor communication;
    5. problems escalated too late for intervention.

Analysis from four common perspectives

Processing the interviews, the failure factors as mentioned were arranged according to type, using the classification of four well-known, often described perspectives. These are:

- organisation ICT strategic perspective (the view of the ICT professional);
- project management perspective (the view of the project manager);
- organisation change perspective (the view of the organisation advisor);
- legal perspective (the view of the lawyer and/or gen. manager).

Some failure factors fall into more than one single category, which makes the choice complex. Nevertheless, a global classification proved to be possible.

Five other aspects

Nico Beenker: "Are the findings of the field study surprising? The answer is: No and Yes. No because the entire list includes many failure factors as already described by Jan Oonincx in 1982, which were later on once more confirmed by many people from the above-described perspectives. And Yes, because during processing of the field study, causality was noticed between the five aspects that did not emerge before within this context. These five aspects as a whole are no part of one of the four professional perspectives as mentioned, which we usually use to look at this problem."

These five aspects (areas of attention) are:

- communication;
- perception;
- fit between supplier and client;

- commercial dynamics;
- type of partnership.

Together these aspects are perhaps the basis for a tool using which success and failure can be more strongly influenced as compared to the usual professional tools. The five aspects are next discussed in brief.

Communication

Communication plays a key part in complex ICT projects. In general, in cases of failure there is too little appreciation and attention to the quality of the (written) communication.

Perception

Many cases of failure are based on differences in perception. The adverse effect of differences in perception usually becomes evident gradually in the project through a deviating evaluation and appreciation of achieved milestones. In that case, these lead to difference of opinion between client and supplier. The supplier has the task to investigate any possible difference in perception at the earliest possible stage and to put these up for discussion. Difference in perception often occur (mentioned by the lawyers in the study) in the fields of risks and preconditions, the contribution of the client's organisation, scope and project progress (scope creep) and organisation and cooperation agreements.

Fit between supplier and client

In cases of failure, there are often one or more existing misfits between supplier and client. For example a misfit between persons with regard to procedures and system development methodologies. It also often happens that suppliers man projects on the basis of availability instead of basing this on the required competencies for the assignment.

Commercial dynamics

Commercial dynamics is the collection of all undesired effects involved in commercial aspects of a valuable task. This could be a supplier who, for instance, consciously leaves certain things undiscussed that are proved to be very relevant at a later stage. There are also clients that knowingly maintain a fundamental unequivalence between themselves and the supplier.

Type of partnership

People often think differently about cooperation. From the point of view of the client, the supplier has to be knowledgeable and supply the system. On the other side are the suppliers who (as things progress) often complain about clients failing to cooperate or their lack of expertise. Causes for failure are for example: no contract or a very concise contract, splitting of control and responsibility and inconsistencies in the contract.

Nico Beenker: "Parallels can be drawn between the manner in which accidents and disasters take place and the way in which complex ICT projects fail. In both cases, small individual events that are connected and amplify each other enable a much larger, disastrous event to take place. The five aspects as described enable us to categorise these events and better monitor these. Therefore, attention from the management for the five aspects is very desirable."

The table below includes the failure factors that were discussed in this paragraph.

Author	No	Description	Priority
		Below a list of the five most often mentioned factors; starting with the factor that is mentioned most often	
NB	01	Planning ahead too optimistic	1
NB	02	Badly phrased contracts	2
NB	03	Poor project management	3
NB	04	Poor communication	4
NB	05	Problems escalated too late	5
		Next five aspects are perhaps the basis for a tool using which success and failure can be more strongly influenced as compared to the usual professional tools	
NB	06	In cases of failure there is too little appreciation and attention to the quality of the (written) communication	
NB	07	Differences in perception	
NB	08	One or more existing misfits between supplier and client	
NB	09	Commercial dynamics (the collection of all undesired effects involved in commercial outsourcing of a valuable task)	
NB	10	Type of partnership. People (supplier and client) often think differently about cooperation	

Table 3.6.3.2: Failure factors in development and introduction of information systems

3.6.4 The present

3.6.4.1 Introduction

In practice, it still proves to be difficult to carry out ICT projects on time and within budget. Peter Noordam et al. [2007]: "However, project management has started on the road to adulthood, as emerges from research in to the factors that are of importance for increasing the chance of success of ICT projects".

3.6.4.2 Failure factors

What exactly makes a project successful? Peter Noordam et al. : "There are many theories on this but in essence, we can say that a successful project satisfies three factors: it complies with the functionality agreed to in advance, it is delivered on time and it is delivered within the agreed budget. When these three factors balance each other, we can speak of a successful project. From various studies over the last few years, it has become clear that delivering the intended functionality on time and within budget is a laborious affair." In a KPMG study amongst 252 controllers [Noordam et al. 2007], the respondents named the following reasons as being the most important ones for the failure of projects:

1. poor project management (32 percent);
2. lack of communication in and around the project (20 percent);
3. objectives not defined (17 percent);
4. unfamiliarity with scope and complexity (17 percent)
5. technical complexity and technical integration issues (7 percent);
6. others (unknown) (7 percent).

Peter Noordam et al. : "We asked ourselves the question what the situation is in 2007, with regard to satisfaction on projects and whether the factors as mentioned are still blamed for the failure of projects. In doing so, we asked ourselves what we could learn from failing and successful ICT projects. We decided to carry out a project management study in order to obtain handles for successful project implementation, based on insight into the success and failure factors of ICT projects."

3.6.4.3 Study design

The study was carried out by Bisnez Management [Noordam et al. 2007] by means of a survey in the period between October and December 2006. The questionnaire was distributed amongst approximately 3000 IT professionals (managers, project managers and IT specialists) in the Netherlands and resulted in over 230 respondents (7.67%). The results of the survey were processed by two teams consisting of students of the Vrije Universiteit Amsterdam and tested against other research data. The interim results were analysed with a selection of the interviewees. Next, during a group session in late December 2006 the results [Business & IT Trend Institute 2007] were presented to and discussed with project managers from different branches.

3.6.4.4 Results

Evaluation ICT projects

Peter Noordam et al. : "Because 'result' is difficult to assess – after all, result involves realisation of the objectives (benefits) of a project as set in advance – for answering the survey we were mainly guided by the *degree of satisfaction with regard to projects*. Satisfaction is more qualitative and reflects a feeling with regard to the project. In the study, the respondents were asked to show their satisfaction with the projects in their organisation by means of a report mark on a scale of between 1 and 10. This resulted in an average mark of 5.5 across the branches. Not a 'nice' average and still a lot of room for improvement". An average mark of 5.5 is insufficient and has to be improved.

Business Case

In order to arrive at decision-making in relation to an ICT project, it is possible to use a business case. This describes the intended objective of the project and provides an outline of the investments required for achieving this objective. The business case, which has to be drawn up by the line manager involved, serves as start of a project. The survey showed that:

- 30 percent of the interviewees do hardly or not all use a business case in connection with their investment decisions;
- 35 percent largely use a business case, in which especially the estimate project costs are filled in correctly;
- 32 percent make full use of a business case.

It is interesting to establish whether the use of a business case contributes to a better result or higher degree of satisfaction with regard to the project. Peter Noordam et al. : "Of all interviewees that indicated to be very dissatisfied with the project, 50% did not draw up a business case and 50% a limited business case. Of all interviewees that indicated to be very satisfied about the project, 100% drew up a complete business case. All this leads to the conclusion that the use of a business case results in a higher degree of satisfaction with the project, whilst the satisfaction with the project is very low when no business case is used. Another interesting detail is also the result that after approval, the business case (drawn up completely or partly) is filed away and never again sees the light of day. This can be concluded from the fact that the question whether the business case is evaluated during or after the project in order to determine whether the expectations were right, it turns out that this only happens in 16 out of 100 projects. The fact whether the costs are evaluated (during and after the project) and whether the benefits are evaluated (at the end of the project) also prove to have a fairly large influence on the success."

Project management methodologies

Over 65 percent of the respondents state that they use a standard project management methodology. Peter Noordam et al. : "It turns out that very dissatisfied respondents do not use a standard

methodology; whilst of the dissatisfied respondents over 45% applies such a methodology. Of the satisfied respondents, more than 75% utilised a standard project management methodology. In general, the satisfaction increases when a standard project management methodology is used (figure 3.4.6.1). It is significant that 55% of the respondents still use a methodology when the project costs less than 75,000 Euros. *This shows that project management is becoming increasingly more mature.*"
Of all used standard methodologies PRINCE2 [Derksen & Noordam 2006] is with 73% by far the most commonly used methodology. With regard to this, it is significant that the biggest users are to be found amongst banks, insurers and the central government. From all banks and insurers, about 85 percent are satisfied to very satisfied about the executed ICT projects. However, at central government there is considerable dissatisfaction with the projects: over half of all respondents are dissatisfied. How to explain this difference? Peter Noordam et al. : "To this purpose, we have looked at another important success factor: *the project size.* Looking at the project size in our study we see significant differences between banks and insurers on the one hand and the central government on the other hand. Very large projects

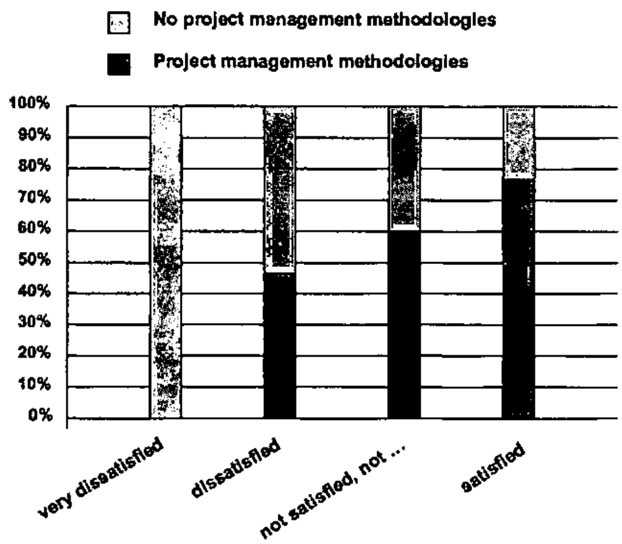


Figure 3.6.4.1: Satisfaction regarding the execution of projects in relation to using or not using standard project management methods.

worth more than 10 million Euros only take place within the central government; 64% of all projects at banks and insurers are worth less than 1.5 million Euros, whilst this is about 43% for the central government. This could explain why the central government is less satisfied as compared to banks and insurers. From the research carried out by the Standish Group International Inc [1998] it can be concluded that it is sensible to set up projects such that these do not exceed the 1.5 million dollar line in order to increase the chance of success (see table 3.6.4.1)."

Project budget ($)	Chance of success (%)
75,000	55
1.5 – 3 million	25
10 million or over	0

Table 3.6.4.1: Project budget versus chance of success project

Quality project manager

Apart from the methodology and project size, the research team has also looked at the project manager as a success factor. In doing so, they looked at three types of projects, with a diminishing degree of technical input:

- *technology projects*: projects in which the technology is the main factor, for example a new operating system or replacement of hardware;
- *integrated projects*: projects in which the technology plays just as important a part as process aspects, for example the implementation of a new information system in which there is significant impact on the (employees within the) organisation;
- *business transformation projects*: projects in which technology plays a secondary part and changing the organisation is the primary objective of the project, for example reorganisations.

Peter Noordam et al. : "Based on the data and the panel discussion it can be stated that there exists no universal project manager who is able to deal equally well with all three types of projects. Albeit that communicative skills are considered important in integrated projects and business transformation projects, in technical projects it will be particularly experience with similar projects that is an important selection factor. It may also be concluded that technical knowledge is certainly an important skill for project managers to have. It may be so that emotional intelligence and functional knowledge are allocated a lot of weight in present-day theory, in practice it does turn out that especially technical knowledge is considered one of the main qualities for concluding projects successfully."

Cyclic risk management

Risk management is inextricably bound up with the execution of projects. Within PRINCE2, the application of risk management is also considered important. There are many conceivable definitions for describing a risk. For this survey, the research team used the following definition: "A risk in the context of a project can be viewed as a factor that may endanger the successful conclusion of a project or may lead to exceeding of costs, exceeding of time and /or qualitative shortcomings". One of the questions within this context was: "When is risk management applied during the execution of the projects within your organisation?

- prior to the start of the project;
- during execution of the project;
- cyclically before and during the execution of the project;
- we do not apply risk management."

Peter Noordam et al. : "From this, it emerges that approximately 26 percent of all respondents applies risk management prior to the start of a project. About 13 percent does this during the project and almost 43 percent applies cyclic risk management, both prior to as well as during the execution of the project. Nevertheless, 18 percent state that they use no risk management at all. When we compare this with the satisfaction, it turns out that cyclic execution of risk management results in the largest amount of satisfaction. An important category of risk concerns the *complexity*. In general, the following emerged:

- technical complexity is still underestimated in ICT projects. This fits in perfectly with the renewed attention to 'architecture' and architect-designed building;
- especially in larger integrated projects it is found that not unexpectedly, the social complexity is still the largest area of risk."

Project evaluation

To conclude, the research team asked whether are evaluated afterwards, with the objective to learn from this for a next project. Peter Noordam: "What is striking is the fact that simply evaluating projects is strongly related to the satisfaction about the successful execution of projects: in a project where evaluation does take place, 80 percent are neutral to satisfied, in a project without evaluation less than balf (44 percent) is neutral to satisfied. An evaluation aimed at having realised the project objectives is mentioned as being the most important type of evaluation. However, according to the interviewees this is still not given enough attention. That also transpires from the question to what extent the benefits are included in the business case and to what extent realisation if these benefits are indeed evaluated. This result favours judging a project not just on whether it has realised the desired functionality on time and within budget, but also favours giving the project responsibility for actually realising the benefits."

Conclusions

ICT projects do still bave a negative image. Too expensive, too late and not achieving the desired functionality is still in 2008 the adage as far as many projects are concerned. Many, especially larger, organisations are aware of this and attempt to secure their experiences and include these in later projects. Now there are a number of developments in which project management seems to grow up and further professionalizes.

The growing attention to business cases and decision-making surrounding ICT projects in general has clearly increased. Peter Noordam et al. : "In our firm belief, 70 percent of all projects should not be allowed to start because insufficient thought was given to the added value of the project for the organisation and the minimal preconditions for making the project successful."

The survey results with regard to the project manager's qualities are surprising. At the evaluation a very experienced project manager, who is now in a position where he trains new project managers stated: "Relatively inexperienced project managers say: '30% of my attention goes to the objectives and 30% goes to the plan of approach, the landmarks and the progress. The rest of my time is spent on the project staff and how they function'. The truly experienced project manager gives the following answer: '10% of my attention is given to planning, objectives et cetera and 90 percent of my time I spend on people'. After all, projects do remain the work of people".

An expected but surprising outcome is the way in which risk management is handled. Peter Noordam et al. : "When risks are given prominent attention at the start but particularly during execution of projects, then this increases the chance of success. Application of this type of risk management is still relative immature and is up until now not much used. By the way, this involves management of risks and not avoiding risks."

The project should not just be started and executed properly but should be primarily evaluated properly. The evaluation results could contribute significantly to the successful execution of new projects.

Peter Noordam et al. : "The gradual introduction of standards and formats that can be useful in the execution of projects is a good thing. This is also part of the professionalisation of the profession. However, slavish and blind imitation of this type of method as a guarantee for success often results in a 'paper tiger' that negatively influences the progress of the project. A number of more experienced organisations that indicate using PRINCE2, also state that they mainly apply PRINCE2 in a pragmatic way instead of following the rules to the letter."
The table below includes the failure factors that were discussed in this paragraph.

Author	No	Description	Priority
		The respondents named the following reasons as being the most important ones for the failure of projects:	
PN	01	Poor project management (KPMG)	1
PN	02	Lack of communication in and around the project (KPMG)	2
PN	03	Objectives not defined (KPMG)	3
PN	04	Unfamiliarity with scope and complexity (KPMG)	4
PN	05	Technical complexity and technical integration issues (KPMG)	5
PN	06	The use of a business case results in a higher degree of satisfaction with the project, whilst the satisfaction with the project is very low when no business case is used	
PN	07	The fact whether the costs are evaluated (during and after the project) and whether the benefits are evaluated (at the end of the project) also prove to have a fairly large influence on the success	
PN	08	The satisfaction increases when a standard project management methodology is used	
PN	09	It is sensible to set up projects such that these do not exceed the 1.5 million dollar line in order to increase the chance of success (Standish Group International Inc.)	
PN	10	Technical knowledge is certainly an important skill for project managers to have	
PN	11	Cyclic execution of risk management results in the largest amount of satisfaction	
PN	12	Technical complexity is still underestimated in IT projects	
PN	13	In larger integrated projects the social complexity is still the largest area of risk	
PN	14	Evaluating projects is strongly related to the satisfaction about the successful execution of projects	
PN	15	This result favours judging a project not just on whether it has realised the desired functionality on time and within budget, but also favours giving the project responsibility for actually realising the benefits	
PN	16	Slavish and blind imitation of project management methods as a guarantee for success often results in a "paper tiger" that negatively influences the progress of the project	

Table 3.6.4.2: Failure factors in development and introduction of information systems

3.6.5 The methods

3.6.5.1 Introduction

Methods for developing information systems (and software) have been receiving attention for a long time. In the Netherlands during 1981 and 1982, a series of no less than 24 articles appeared in the magazine Informatie. One of these articles is "De methode doet het niet" (The method does not do it) by Jaap van Rees [1982]. Amongst other things, he draws the conclusion that: "It is the designer who designs, not the method". Tom Gilb [1988] is also takes up a critical position towards the method: "Don't believe blindly in any one method; use your methods and common sense to measure the reality against your needs". In November 2007, the article "Projectmanagement is risicomanagement" (Project management is risk management) [Molendijk & Oud 2007] was published. In paragraph 3.6.5.3, this is discussed further.

3.6.5.2 Methods used

Methods often used for developing information systems/software are [Molendijk & Oud 2007, Neering et al. 2006, Dalcher & Brodie 2007, Gilb & Brodie 2005]:

SDM (Software Development Methodology)
A concrete interpretation of the waterfall approach which is particularly known in the Netherlands. In a waterfall approach, development phases – from the definition study to the implementation – happen sequentially.

DSDM (Dynamic Systems Development Method)
An approach based on a strong degree of user participation, iterative development, self organising teams and an informal approach. Is often associated with techniques such as prototyping and time boxing.

RUP (Rational Unified Process)
A framework offered through a tool for creating a project specific approach based upon amongst other things, iterative architecture-oriented development. Is characterised by its completeness and the strong relation to UML (use cases).

XP (Extreme Programming)
The best known 'agile' method. Known for its short iterations, focus on simplicity, face-to-face communication, a fully participating customer and for 'practices'.

Prince 2
Almost three-quarters of the project management methods prove to be based on Prince 2 [Molendijk & Oud 2007].

Of all above system development methods, the methods that are based on DSDM prove to be the most successful [Neering et al. 2006].

EVO (Evolutionary project management / delivery)
EVO also supported the related concept that projects do not initially have the 'final and correct user requirements' specified. The underlying principle of EVO is the Plan-Do-Study-Act cycle. The new

system is delivered in a series of small steps.

Other used methods are for example:

- MSF (Microsoft Solution Framework);
- Scrum;
- RAD (Rapid Application Development);
- IAF (Integrated Architecture Framework);
- TOGAF (The Open Group Architecture Framework).

3.6.5.3 Project management is risk management

"Selection of a particular software development method has major consequences for the management of a project". In the article "Projectmanagement is risicomanagement" (Project management is risk management) Klaas-Jan Molendijk and Stef Oud [2007] of Deloitte discuss this proposition. An outline follows below. The Success/Failure factors in ICT projects derived from this are included in table 3.6.5.1.

Outline

- Success or failure of a software project is all about management;
- By choosing a particular method as the starting position of a project means that we make many implicit management choices. An 'agile' method for instance, does not combine with an authoritarian management style;
- One method develops everything in one single go, whilst another divides the software into smaller, easier to manage projects (iterations);
- One method describes in detail 'what' the project members have to do and 'how' they should do this, whilst another method leaves the project team considerable freedom of choice. In XP the choice is for working primarily on the basis of 'tacit knowledge' (knowledge existing in people's heads) as opposed to documented knowledge, does function well in the case of 'moving' functional requirements. This means that changes can be implemented faster and easier. However, the power of XP is considerably less when used with a long-term project because of the loss of knowledge that is risked through inevitable staff turnover;
- Every project is different and therefore demands a different approach. When the demands are clear and stable, we choose a waterfall approach. When large dynamics are involved and software with a high degree of interactivity, we choose an 'agile' method such as XP or Scrum;
- Because there are projects risks that we wish to avoid, we make certain choices. Therefore, software project management is in essence risk management;

- One method is better at preventing/limiting risk than another;
- Experience teaches us that a project continually encounters problems (the occurrence of risks) and that up to the end, the initial approach evolves into the 'right approach';
- It is known that we are not capable of defining and weighing up all relevant factors (risks);
- Software project management is complex. Many software projects are too reactive in nature: as soon as anything goes wrong, this is pointed out and action is taken. However, the problem is that by that time costs have already been made, time has been lost and frustrations created;
- Selection of a different method is not always the solution but different application of a method may be;
- There is no check on methods and therefore there is also no check on the quality of software development.

Author	No	Description
JRR	01	The method does not work
JRR	02	The designer designs and not the method
TG	01	Don't believe blindly in any one method; use your methods and common sense to measure the reality against your needs
KJM	01	Success or failure of a software project is all about management
KJM	02	Selection of a particular software development method has major consequences for the management of a project
KJM	03	Selection of a different method is not always the solution but different application of a method is
KJM	04	Project management is risk management

Table 3.6.5.1: Success / Failure factors when using methods

3.7 COHERENCE

3.7.1 Introduction

In this section, we examine the question: What do others think of ICT projects failing or not failing? Various authors describe the results of research into success and failure factors in development and introduction of information systems. Some authors have also published books on this subject [for example Smith 2001, Oonincx 1982, Jones 1996-2]. For that reason, this collection of publications may also be considered as representative for this subject as far as this thesis is concerned.

Definitions

It may be concluded that the definitions as provided by the various authors are not univocal. Some authors are of the opinion that a project fails when it does not achieve successful implementation, other authors take this further and include the success of the information system in the organisation in relation with the user satisfaction and the benefits for the business in their assessment. In Chapter 2, a choice was made concerning the definition as used within this thesis. For this thesis a project failure has one or more of the following characteristics:

1. it does not comply with the functionality agreed to in advance, including agreed changes of scope;
2. it exceeds the planned time-scale by more than 50%, excluding the time-scale impact of agreed changes in scope;
3. it exceeds the build cost by more than 50%, excluding the cost of agreed changes in scope.

3.7.2 Results

The results of the various studies are shown in the following tables:

- table 3.7.1a and table 3.7.1b [John Smith];
- table 3.7.2 [Capers Jones];
- table 3.7.3 [Michael Evans et al.];
- table 3.7.4 [K.T. Yeo];
- table 3.7.5 [Lorin May];
- table 3.7.6 [John S. Reel];
- table 3.7.7 [Jan Oonincx];
- table 3.7.8 [Nico Beenker];
- table 3.7.9 [Noordam et al.];
- table 3.7.10 [Methods].

It is obvious that some success and failure factors are brought forward by several authors. By including these duplicates only once in the results, a collection of success and failure factors is created that may be used as a frame of reference for the next chapters of this thesis. This is shown in figure 3.7.1 (input -> process -> output). In this process of removing the superfluous specimens of the duplicates the following steps are taken:

Step 1

It is an interesting fact that, after analysing several sources of information as well as his own experiences, John Smith found 40 *generic* root causes of troubled projects [Smith 2001]. His list is shown in table 3.7.1a and 3.7.1b. For that reason, this collection of success and failure factors is used for a basis.

Step 2

The above-mentioned tables are used as a point of departure.

Step 3

At every success/failure factor that can be viewed as a duplicate of a success/failure factor that appears in a table with a *lower* number, the table index in question is stated. Example: at LM 09 "Poor architecture" the column "Related to" states: JS RC19 (LM = Lorin May; JS= John Smith). Within a table, a duplicate may sometimes appear. In that case, it is also possible to include a table index in the column "Related to". Circle references are not permitted.

Step 4

After step 3 has been fully executed all success/failure factors that are provided with a "Related to" will disappear.

Step 5

The success/failure factors of John Smith will end up in result table 3.7.11.
The other remaining success/failure factors will come under result table 3.7.12.

Step 6

Tables 3.7.11 and 3.7.12 form, together with the success/failure factors of Tarek Abdel-Hamid (paragraph 3.8) and the "Big hitters" (chapter 4) the frame of reference or the reference model success/failure factors.

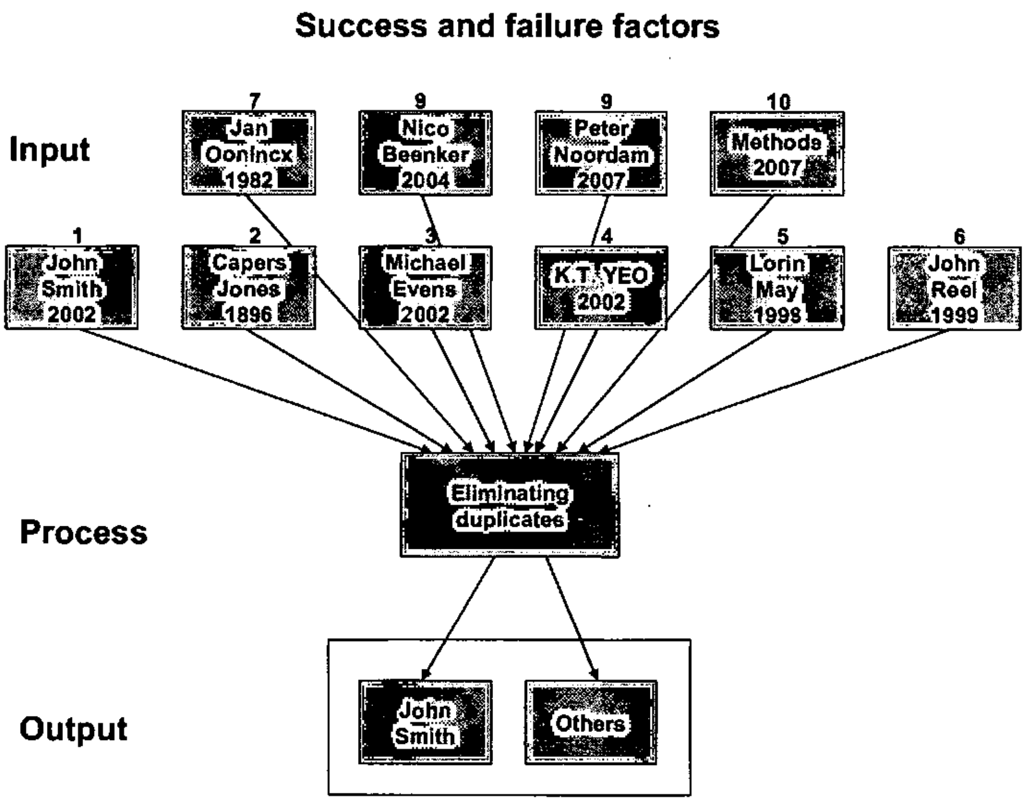


Figure 3.7.1: The "Eliminating duplicates" process: removing the superfluous specimens of the duplicates

John Smith, 2002

Author	No	Description
Project conception		
JS	RC01*	Project based on an unsound premise or an unrealistic business case
JS	RC02*	Buyer failure to define clear project objectives, anticipated benefits and success criteria
JS	RC03*	Project based on state-of-the-art and immature technology
JS	RC04*	Lack of buyer board-level ownership/commitment or competence
JS	RC05*	Buyer's funding and/or time-scale expectations unrealistically low
JS	RC06*	Buyer failure to break a complex project into phases or smaller projects
Project initiation/mobilisation		
JS	RC07*	Vendor setting unrealistic expectations on cost, time-scale or vendor capability
JS	RC08*	Buyer failure to define and document requirements (functiooal and non-functional)
JS	RC09*	Failure to achieve an open, robust and equitable buyer-vendor relationship
JS	RC10*	Vendor failure to invest enough resources to scope the project to contract
JS	RC11*	Buyer lack of sufficient involvement of eventual end-users
JS	RC12*	Vendor uoderestimation of resources (predominantly person-effort) required
JS	RC13	Vendor failure to define project tasks, deliverables and acceptance processes
JS	RC14*	Failure to actively manage risks and maintain robust contingency plans
JS	RC15*	Poor project planning, management and execution
JS	RC16	Failure to clearly define roles and responsibilities in the contract/subcontracts
JS	RC17	Full-scope, fixed-price contracting (requirements, design and developmeot)
System design		
JS	RC18*	Failure to 'freeze' the requirements baseline and apply change control
JS	RC19*	Poor choice of technical platform and/or architecture
JS	RC20	Vendor starting a phase prior to completing a previous phase
JS	RC21*	Poor choice of design/development method
JS	RC22*	Failure to undertake effective project reviews and take decisive action
JS	RC23*	Vendor lack/loss of skilled resources
JS	RC24	Poor vendor standards deployment (design, coding, testing, configuration management, etc.)
JS	RC25*	Poor vendor requirements traceability (requirements > design > code > test)
JS	RC26	Buyer retention of design authority with right to approve/reject low-level designs
System development		
JS	RC27	Delays causes the project to be overtaken by advances in technology
JS	RC28	Vendor failure to 'freeze' the design (and technical platform) and apply change control
JS	RC29	Inadequate vendor training and supervision of junior staff
JS	RC30	Inadequate vendor review of designs/code/documentation
JS	RC31	Poor vendor management of sub-contractors
JS	RC32	Lack of a formal, 'engineering' approach to integration and testing by vendor
JS	RC33*	Insufficient attention paid by vendor to non-functional requirements
System implementation		
JS	RC34*	Buyer failure to manage the change implicit in the project (people, processes, technology)
JS	RC35	Inadequate user/systems training
JS	RC36	Catastrophic failure of the system, with no effective contingency arrangement
JS	RC37	Missing a crucial 'go live' date
System operation, benefit delivery, stewardship and disposal		
JS	RC38*	Buyer failure to measure actual delivered benefit and take corrective action
JS	RC39	Buyer failure to maintain/enhance system post-implementation
JS	RC40	Changes in the competitive or macro-economic environment

*) is also mentioned by another researcher

Table 3.7.1a: Root causes (RC) of troubled projects [John Smith]

John Smith, 2002

Author	No	Description
JS	PUBRC01*	Lack of senior management involvement and commitment
JS	PUBRC02*	Failure to focus on key business and end-user needs
JS	PUBRC03	Failure to break complex projects into manageable, separately contracted 'cbunks'
JS	PUBRC04	Poor and unimaginative project management
JS	PUBRC05	Poor risk management and contingency planning
JS	PUBRC06*	Unclear contracts and poor contract management
JS	PUBRC07	Insufficient focus on user training needs and the design of training interventions

*) is also mentioned by another researcher

Table 3.7.1b: Root causes (RC) of troubled projects in the UK public sector [John Smith]

Capers Jones, 1996

Author	No	Description	Related to	
		Large software systems (FP = Function Point)		
CJ	01	1.000 FP's: Quality control is a major requirement at this size range		
CJ	02	1.000 FP's: With team development (up to 10 staff memhers), issues of system segmentation and interfaces among components become problematic		
CJ	03	10.000 FP's: This size range is often plagued by cost and schedule overruns and by outright cancellations		
CJ	04	10.000 FP's: Development teams of 100 or so are common, so communication and interface problems are endemic		
CJ	05	10.000 FP's: Configuration control and changc management are mandatory for this size plateau		
CJ	06	100.000 FP's: Development teams number in the bundreds, often in multiple locations that may even be in different countries. Communication problems are rampant		
CJ	07	100.000 FP's: Formal configuration control and change management are mandatory and expensive for this size plateau		
CJ	08	A surprisingly strong influence of the outcome of software projects is the nature of the industry that builds the application		
CJ	09	The systems software community has learned the hard way that careful quality control is on the critical path		
CJ	10	Military standards are so complex that the productivity of defence application development projects is lower than in any other software sub industry		
CJ	11	The MIS (management information systems) community lags in quality control and testing technologies compared to the other communities		
CJ	12	As the overall size ranges grow larger, delays and cancellations become more common and also more severe		
CJ	13	None of the (six) domains has fully mastered the ability to construct truly large software systems without a significant risk of termination or cancellation		
CJ	14	No organisation should tackle software projects above 10.000 FP's without fully evaluating schedules, costs, risks and value		
CJ	15	The probability of recovery for a software project in deep distress		

		is fairly low. Prevention is often more effective than control		
CJ	16	Successful projects deploy a notable quantity of quality control and project management automation, but tools by themselves do not make successful projects. Capable mangers and capable technical personnel are also needed		

Table 3.7.2: Success / failure factors [Capers Jones]

Evans et al., 2002

Author	No	Description	Related to	
		Seven characteristics of dysfunctional software projects		
EV	01	Failure to apply essential project management practices		
EV	02	Unwarranted optimism and unrealistic management expectations	JS	RC05
EV	03	Failure to implement effective software processes		
EV	04	Premature victory declarations		
EV	05	Lack of program management leadership	JS	RC15
EV	06	Untimely decision-making		
EV	07	Lack of proactive risk management		

Table 3.7.3: Success / failure factors [Michael Evans et al.]

K.T. Yeo, 2002

Author	No	Description	Related to	
		Critical failure factors in information system projects		
		Process driven issues (business planning, project planning, project management and control)		
KY	01	Underestimate of timeline	JS	RC07
KY	02	Weak definitions of requirements and scope		
KY	03	Inadequate project risk analysis		
KY	04	Incorrect assumptions regarding risk analysis		
KY	05	Ambiguous business needs and unclear vision	JS	RC01,RC08
		Context driven issues(corparate culture, corparate management, users, politics)		
KY	06	Lack user involvement and inputs from the onset	JS	RC11
KY	07	Top down management style		
KY	08	Poor internal communication		
KY	09	Absence of an influential champion and change agent		
KY	10	Reactive and not pro-active in dealing with problems		
		Content driven issues (information technology, business process and system design, IT/IS professional and knowledge sources)		
KY	11	Consultant/vendor underestimated the project scope and complexity	JS	RC10
KY	12	Incomplete specifications when project started		
KY	13	Inappropriate choice of software		
KY	14	Changes in design specifications late the project	JS	RC18
KY	15	Involve high degree of customisation in application		

Table 3.7.4: Success / failure factors [K.T. Yeo]

Lorin May, 1998

Author	No	Description	Related to	
		The failure causes are listed in no particular order		
LM	01	Poor user input	JS	RC02,RC11
LM	02	Stakeholder conflicts		
LM	03	Vague requirements	JS	RC08
LM	04	Poor cost and schedule estimation	JS	RC05,RC07,RC12
LM	05	Skills that do not match the job	JS	RC23
LM	06	Hidden costs of going "lean and mean"		
LM	07	Failure to plan	JS	RC15
LM	08	Communication breakdowns		
LM	09	Poor architecture	JS	RC19
LM	10	Late failure warning signals		

Table 3.7.5: Success / failure factors [Lorin May]

John S. Reel, 1999

Author	No	Description	Related to	
		The five essential/critical factors to managing a successful software project		
JR	01	Start on the right foot (nr 06 -15)	JR	06 - 15
JR	02	Maintain momentum (attrition, quality and management)		
JR	03	Track progress (a large problem in managing software development is figuring where you are in your schedule)	JS	RC25
JR	04	Make smart decisions (always use commercial libraries when available)		
JR	05	Institutionalise post-mortem analyses (if you do not take time to figure out what happened during a project, both the good and the bad, you are doomed to repeat it)	JS	RC22
		Tom Field [55] gave 10 signs of IS project failure		
JR	06	Project managers don't understand users' needs	JS	RC08,RC33,PUBRC02
JR	07	The project's scope is ill-defined	JS	RC02,RC08
JR	08	Project changes are managed poorly	JS	RC34
JR	09	The chosen technology changes	JS	RC19
JR	10	Business needs change	JS	RC34
JR	11	Deadlines are unrealistic	JS	RC01,RC05,RC07
JR	12	Users are resistant	JS	RC11
JR	13	Sponsorship is lost	JS	RC04,PUBRC01
JR	14	The project lacks people with appropriate skills	JS	RC23
JR	15	Managers ignore best practices and lessons learned	JS	RC01

Table 3.7.6: Success / failure factors [John Reel]

Jan Ooninck, 1982

Author	No	Description	Related to	
JO	01	The lack of a methodology for developing information systems	JS	RC21
JO	02	Insufficient insight into decision-making processes and information requirements	JS	RC08
JO	03	The feasibility of integral information systems	JS	RC06
JO	04	Insufficient involvement and cooperation of users in setting up information systems	JS	RC11
JO	05	Insufficient and/or inefficient use of the possibilities as offered by the computer in the information system		
JO	06	Not using the new information system		
JO	07	Collection and storage of too much and unsuitable data		
JO	08	Ineffectiveness and limited applications of traditional systems for data storage	JS	RC19
JO	09	Set-up of information systems too static		
JO	10	Insufficient suitable human resources for development and implementation of information systems	JS	RC23
JO	11	The influence of changes to the information system on the organisation	JS	RC34
JO	12	Too little attention to the informal aspects in relation to the information system		
JO	13	The development process takes too long and the costs are too high		
JO	14	Computerisation based on status considerations		
JO	15	Too little attention to the aspect of profitability and the lack of measuring instruments for determining the benefits of the information system	JS	RC38
JO	16	Unsuitable equipment and software	JS	RC03
JO	17	Insufficient attention to reliability and controllability		

Table 3.7.7: Success / failure factors [Jan Ooninck]

Nico Beenker, 2004

Author	No	Description	Related to	
NB	01	Planning ahead too optimistic	JS	RC01,RC05
NB	02	Badly phrased contracts	JS	PUBRC06
NB	03	Poor project management	JS	RC15
NB	04	Poor communication	NB	06
NB	05	Problems escalated too late		
		Next five aspects are perhaps the basis for a tool using which success and failure can be more strongly influenced as compared to the usual professional tools		
NB	06	In cases of failure there is too little appreciation and attention to the quality of the (written) communication		
NB	07	Differences in perception		
NB	08	One or more existing misfits between supplier and client	JS	RC07
NB	09	Commercial dynamics (the collection of all undesired effects involved in commercial outsourcing of a valuable task)	JS	RC09
NB	10	Type of partnership. People (supplier and client) often think differently about cooperation	JS	RC09

Table 3.7.8: Success / failure factors [Nico Beenker]

Peter Noordam et al., 2007

Author	No	Description	Related to	
-				
PN	01	Poor project management (KPMG)	JS	RC15
PN	02	Lack of communication in and around the project (KPMG)	NB	06
PN	03	Objectives not defined (KPMG)	JS	RC02
PN	04	Unfamiliarity with scope and complexity (KPMG)		
PN	05	Technical complexity and technical integration issues (KPMG)		
PN	06	The use of a business case results in a higher degree of satisfaction with the project, whilst the satisfaction with the project is very low when no business case is used		
PN	07	The fact whether the costs are evaluated (during and after the project) and whether the benefits are evaluated (at the end of the project) also prove to have a fairly large influence on the success		
PN	08	The satisfaction increases when a standard project management methodology is used		
PN	09	It is sensible to set up projects such that these do not exceed the 1.5 million dollar line in order to increase the chance of success (Standish Group International Inc.)	JS	RC06
PN	10	Technical knowledge is certainly an important skill for project managers to have		
PN	11	Cyclic execution of risk management results in the largest amount of satisfaction	JS	RC14
PN	12	Technical complexity is still underestimated in IT projects	PN	05
PN	13	In larger integrated projects the social complexity is still the largest area of risk		
PN	14	Evaluating projects is strongly related to the satisfaction about the successful execution of projects		
PN	15	The result favours judging a project not just on whether it has realised the desired functionality on time and within budget, but also favours giving the project responsibility for actually realising the benefits		
PN	16	Slavish and blind imitation of project management methods as a guarantee for success often results in a "paper tiger" that negatively influences the progress of the project		

Table 3.7.9: Success / failure factors [Noordam et al.]

Methods, 2007

Author	No	Description	Related to	
JRR	01	The method does not work	TG	01
JRR	02	The designer designs and not the method		
TG	01	Don't believe blindly in any one method; use your methods and common sense to measure the reality against your needs		
KJM	01	Success or failure of a software project is all about management	JS	RC15
KJM	02	Selection of a particular software development method has major consequences for the management of a project	JS	RC21
KJM	03	Selection of a different method is not always the solution but different application of a method is	JS	RC21
KJM	04	Project management is risk management	JS	RC15

Table 3.7.10: Success / Failure factors when using methods [Van Rees 1982, Molendijk & Oud 2007, Neering et al. 2006]

John Smith, 2002

Author	No	Description
Project conception		
JS	RC01*	Project based on an unsound premise or an unrealistic business case
JS	RC02*	Buyer failure to define clear project objectives, anticipated benefits and success criteria
JS	RC03*	Project based on state-of-the-art and immature technology
JS	RC04*	Lack of buyer board-level ownership/commitment or competence
JS	RC05*	Buyer's funding and/or time-scale expectations unrealistically low
JS	RC06*	Buyer failure to break a complex project into phases or smaller projects
Project initiation/mobilisation		
JS	RC07*	Vendor setting unrealistic expectations on cost, time-scale or vendor capability
JS	RC08*	Buyer failure to define and document requirements (functional and non-functional)
JS	RC09*	Failure to achieve an open, robust and equitable buyer-vendor relationship
JS	RC10*	Vendor failure to invest enough resources to scope the project to contract
JS	RC11*	Buyer lack of sufficient involvement of eventual end-users
JS	RC12*	Vendor underestimation of resources (predominantly person-effort) required
JS	RC13	Vendor failure to define project tasks, deliverables and acceptance processes
JS	RC14*	Failure to actively manage risks and maintain robust contingency plans
JS	RC15*	Poor project planning, management and execution
JS	RC16	Failure to clearly define roles and responsibilities in the contract/subcontracts
JS	RC17	Full-scope, fixed-price contracting (requirements, design and development)
System design		
JS	RC18*	Failure to 'freeze' the requirements baseline and apply change control
JS	RC19*	Poor choice of technical platform and/or architecture
JS	RC20	Vendor starting a phase prior to completing a previous phase
JS	RC21*	Poor choice of design/development method
JS	RC22*	Failure to undertake effective project reviews and take decisive action
JS	RC23*	Vendor lack/loss of skilled resources
JS	RC24	Poor vendor standards deployment (design, coding, testing, configuration management, etc.)
JS	RC25*	Poor vendor requirements traceability (requirements > design > code > test)
JS	RC26	Buyer retention of design authority with right to approve/reject low-level designs
System development		
JS	RC27	Delays causes the project to be overtaken by advances in technology
JS	RC28	Vendor failure to 'freeze' the design (and technical platform) and apply change control
JS	RC29	Inadequate vendor training and supervision of junior staff
JS	RC30	Inadequate vendor review of designs/code/documentation
JS	RC31	Poor vendor management of sub-contractors
JS	RC32	Lack of a formal, 'engineering' approach to integration and testing by vendor
JS	RC33*	Insufficient attention paid by vendor to non-functional requirements
System implementation		
JS	RC34*	Buyer failure to manage the change implicit in the project (people, processes, technology)
JS	RC35	Inadequate user/systems training
JS	RC36	Catastrophic failure of the system, with no effective contingency arrangement
JS	RC37	Missing a crucial 'go live' date
System operation, benefit delivery, stewardship and disposal		
JS	RC38*	Buyer failure to measure actual delivered benefit and take corrective action
JS	RC39	Buyer failure to maintain/enhance system post-implementation
JS	RC40	Changes in the competitive or macro-economic environment
Author	**No**	**Description (UK public sector)**
JS	PUBRC01*	Lack of senior management involvement and commitment
JS	PUBRC02*	Failure to focus on key business and end-user needs

JS	PUBRC03	Failure to break complex projects into manageable, separately contracted 'chunks'
JS	PUBRC04	Poor and unimaginative project management
JS	PUBRC05	Poor risk management and contingency planning
JS	PUBRC06*	Unclear contracts and poor contract management
JS	PUBRC07	Insufficient focus on user training needs and the design of training interventions

*) is also mentioned by another researcher

Table 3.7.11: Root causes (RC) of troubled projects [John Smith]

Other authors

Author	No	Description
		Large software systems (FP = Function Point)
CJ	01	1.000 FP's: Quality control is a major requirement at this size range
CJ	02	1.000 FP's: With team development (up to 10 staff members), issues of system segmentation and interfaces among components become problematic
CJ	03	10.000 FP's: This size range is often plagued by cost and schedule overruns and by outright cancellations
CJ	04	10.000 FP's: Development teams of 100 or so are common, so communication and interface problems are endemic
CJ	05	10.000 FP's: Configuration control and change management are mandatory for this size plateau
CJ	06	100.000 FP's: Development teams number in the hundreds, often in multiple locations that may even be in different countries. Communication problems are rampant
CJ	07	100.000 FP's: Formal configuration control and change management are mandatory and expensive for this size plateau
CJ	08	A surprisingly strong influence of the outcome of software projects is the nature of the industry that builds the application
CJ	09	The systems software community has learned the hard way that careful quality control is on the critical path
CJ	10	Military standards are so complex that the productivity of defence application development projects is lower than in any other software sub industry
CJ	11	The MIS (management information systems) community lags in quality control and testing technologies compared to the other communities
CJ	12	As the overall size ranges grow larger, delays and cancellations become more common and also more severe
CJ	13	None of the (six) domains has fully mastered the ability to construct truly large software systems without a significant risk of termination or cancellation
CJ	14	No organisation should tackle software projects above 10.000 FP's without fully evaluating schedules, costs, risks and value
CJ	15	The probability of recovery for a software project in deep distress is fairly low. Prevention is often more effective than control
CJ	16	Successful projects deploy a notable quantity of quality control and project management automation, but tools by themselves do not make successful projects. Capable mangers and capable technical personnel are also needed
EV	01	Failure to apply essential project management practices
EV	03	Failure to implement effective software processes
EV	04	Premature victory declarations
EV	06	Untimely decision-making
EV	07	Lack of proactive risk management
KY	02	Weak definitions of requirements and scope
KY	03	Inadequate project risk analysis
KY	04	Incorrect assumptions regarding risk analysis

KY	07	Top down management style
KY	08	Poor internal communication
KY	09	Absence of an influential champion and change agent
KY	10	Reactive and not pro-active in dealing with problems
KY	12	Incomplete specifications when project started
KY	13	Inappropriate choice of software
KY	15	Involve high degree of customisation in application
LM	02	Stakeholder conflicts
LM	06	Hidden costs of going "lean and mean"
LM	08	Communication breakdowns
LM	10	Late failure warning signals
JR	02	Maintain momentum (attrition, quality and management)
JR	04	Make smart decisions (always use commercial libraries when available)
JRR	02	The designer designs and not the method
JO	05	Insufficient and/or inefficient use of the possibilities as offered by the computer in the information system
JO	06	Not using the new information system
JO	07	Collection and storage of too much and unsuitable data
JO	09	Set-up of information systems too static
JO	12	Too little attention to the informal aspects in relation to the information system
JO	13	The development process takes too long and the costs are too high
JO	14	Computerisation based on status considerations
JO	17	Insufficient attention to reliability and controllability
NB	05	Problems escalated too late
NB	06	In cases of failure there is too little appreciation and attention to the quality of the (written) communication
NB	07	Differences in perception
PN	04	Unfamiliarity with scope and complexity (KPMG)
PN	05	Technical complexity and technical integration issues (KPMG)
PN	06	The use of a business case results in a higher degree of satisfaction with the project, whilst the satisfaction with the project is very low when no business case is used
PN	07	The fact whether the costs are evaluated (during and after the project) and whether the benefits are evaluated (at the end of the project) also prove to have a fairly large influence on the success
PN	08	The satisfaction increases when a standard project management methodology is used
PN	10	Technical knowledge is certainly an important skill for project managers to have
PN	13	In larger integrated projects the social complexity is still the largest area of risk
PN	14	Evaluating projects is strongly related to the satisfaction about the successful execution of projects
PN	15	The result favours judging a project not just on whether it has realised the desired functionality on time and within budget, but also favours giving the project responsibility for actually realising the benefits
PN	16	Slavish and blind imitation of project management methods as a guarantee for success often results in a "paper tiger" that negatively influences the progress of the project
TG	01	Don't believe blindly in any one method; use your methods and common sense to measure the reality against your needs

Table 3.7.12: Success/failure factors in ICT projects [several authors]

3.8 SOFTWARE PROJECT DYNAMICS – AN INTEGRATED APPROACH

3.8.1 Introduction

This section discusses *"the procedures in Tarek Abdel-Hamid's work on Software Project Management"* [Abdel-Hamid & Madnick 1991]. For each chapter the main subjects of *"the procedures in Tarek Abdel-Hamid's work on Software Project Management"* have been mapped point by point in appendix 2. Based on this information, SUFFIs are derived in this paragraph. These are shown in table 3.8.1. Literature references are stated with regard to a number of SUFFIs. The SUFFIs in table 3.8.1 are also compared with the SUFFIs in the tables 3.7.11 (John Smith) and 3.7.12 (others). Any possible relations are indicated as "related to". Remarks are made with regard to some SUFFIs. The SUFFIs in table 3.8.1 are subdivided into two categories, namely: category A and category B. In my opinion, the SUFFIs in category A are the main SUFFIs and these are therefore included in table 3.8.2. In my opinion, the SUFFIs in category B are also interesting to (potential) project managers. For that reason, these SUFFIs are included in table 3.8.3. In paragraph 3.8.5 the conclusions come up for discussion. The SUFFI model is introduced in paragraph 3.8.6.

3.8.2 SUFFI table TAH

Tarek Abdel-Hamid and Stuart E. Madnick, 1991

Au-thor	No	Description	Related to		
Introduction					
TAH	01	Software engineering encompasses both the technical aspects of software development as well as the managerial ones {33,47,246} *)	B	JS	RC15
TAH	02	Poor management can increase software costs more rapidly than any other factor {261}	B	JS	RC15
TAH	03	We still lack the fundamental understanding of the software development process {87,96,173} and without such an understanding the likelihood of any significant gains in the management of software development front is questionable {36,130,161}	B	EV TAH	03 42
TAH	04	There are hundreds of variables that affect software development. Furthermore, these variables are not independent; many of them are related to one another {119,189}. THA constructed *a holistic model* of the software development process	B	TAH	42
TAH	05	The behaviour of an individual subsystem in isolation may be different from its behaviour when it interacts with other subsystems {68}	B	AvD JS	Remark 01 RC32
TAH	06	Brooks' Law: adding more people to a late software project makes it later {57} [Brooks 1995]	B	TAH AvD	64 Remark 02
TAH	07	People under time pressure don't work better, they just work faster. ... In the struggle to deliver any software at all, the first casualty has been consideration of the quality of the software delivered {81}	B	JS	RC15

TAH	08	Two factors significantly influence the initial estimate of the job size: (1) the firm's previous experience; and (2) the general over-all tendency to underestimate the job size {49,223}	B	JS JS	RC05 RC15
TAH	09	System complexity grows as the square of the number of system elements; therefore, experience with a small system cannot account for all the things that will bave to be done in a large system. Neither will the Experience Method apply to systems of totally different content {23}	A	AvD PN	Remark 03 05
TAH	10	In a multi-project environment, competition for company resources becomes a significant dimension	B	AvD JS	Remark 04 RC15
TAH	11	It is difficult to measure performance in programming {181}	B	AvD JS TAH TAH	Remark 05 RC15 33 34
TAH	12	The longer an error goes undetected, the more extensive the necessary rework and the greater the cost	A	TAH TAH	31 32
TAH	13	Inadequate planning is the primary reason for loss of control on many computer programming projects {208}	B	JS	RC15
TAH	14	The relationship between cost and system size is not linear. In fact, cost increases approximately exponentially as size increases {188}	A	AvD	Remark 03
TAH	15	Frequent budget manipulations by management to avoid overruns make historical cost data questionable {188}	B	TAH	58
TAH	16	A major challenge to managers is to motivate employees. A mechanism for motivation, whicb is attracting interest in the software engineering field, is "goal setting" {49}	A	PN	06
TAH	17	All programmers are optimists. They always assume that "This time it will surely run" or "I just found the last bug" {57}	B	JS	RC15
Human Resource Management					
TAH	18	Newly hired project members pass through an orientation during which they are less than fully productive {75,243}	A	JS	RC15
TAH	19	The training of newcomers, both technical and social, is usually carried out by the "old-timers". This is costly because "wbile the old-timer is helping the new employee learn the job, his own productivity on his other work is reduced {55,74,77}	B	TAH JS	18 RC15
TAH	20	Toward the end of the project there is likely to be reluctance to bring in new people	B	TAH	64
Software production / development					
TAH	21	Under schedule pressures, walk throughs and inspections are usually the greatest casualties, they are not only relaxed but often suspended altogether {94}	B	TAH TAH	75 76
TAH	22	Problems of communication and motivation are responsible for inadequacies in process and for consequent losses in productivity {240}	B	KY TAH	08 16
TAH	23	Work force experience level and increases in project familiarity due to learning-curve effects affect tbe productivity of a software development project {76,234,261}	B	JS	RC15
TAH	24	The project goals and schedules can play a significant motivational role throughout the life of a software development project	B	JS PN TAH	RC02 06 16

TAH	25	When a project is perceived to be behind schedule, people tend to work harder to bring it back on schedule. Tbere is a thresbold beyond whicb employees would not be willing to work at an "above-normal" rate {169}	A	JS PN	RC15 07
TAH	26	When project members perceive some "excesses" in the scbedule (the case of negative scbedule pressure), some excesses will be "absorbed" by the workers as "under-work" before downward adjustments are made in the project's scbedule {49,131}. As with positive schedule pressures, there are limits on bow in the case of negative scbedule pressure employees are willing or allowed to absorb	B	JS	RC15
TAH	27	Communication overhead increases in proportion to n^2, where n is the size of the team {57,180,231,236,273}	B	KY	08
TAH	28	Design errors in the early design pbase are generated at a higher rate than are coding errors {165}	B	TAH	32
TAH	29	Newly hired employees are not only less productive on average but also more error-prone than their experienced counterparts {92,189}	B	TAH JS	18 RC14
TAH	30	Schedule pressures often result in the "overlapping of activities that would have been accomplished better sequentially", and overlapping can significantly increase the cbance of errors {252}	A	JS JS	RC15 RC14
System testing					
TAH	31	We assume that undetected errors will become either "Active errors" (errors that produces more errors) or "Passive errors". Because design specs are the blueprints of the system's code, any errors in design will get translated into coding errors	A	TAH TAH	12 32
TAH	32	The earlier the undetected error is, the more "generations" of errors it will produce, and thus the more costly it will end up being	B	TAH	12
Controlling / planning					
TAH	33	Progress in a software project is measured by the number of resources consumed, tasks completed, or both	B	JS TAH	RC15 11
TAH	34	Progress, especially in the earlier phases of software development, is measured by the rate of expenditure of resources rather than by some count of accomplishments {81}	B	JS TAH	RC15 11
TAH	35	As a software project develops, project members often realise that they have under-estimated the number of tasks that constitutes the software system being developed {61}	B	JS	RC15
TAH	36	Tbere is a powerful tendency to focus on the highly visible mainline components of the software, and to underestimate or completely miss the unobtrusive components (e.g., help message processing, error processing, and moving data around {49}	B	JS	RC15
TAH	37	In some cases the management becomes increasingly willing to "pay any price" necessary to avoid oversbooting the "Maximum Tolerable Completion Date". In sucb cases, management is often willing to hire more people	B	TAH	64
Case (the NASA DE-A Software Project)					
TAH	38	At tbe System Development Section 25% of an experienced employee's time is committed per new employee	B	TAH	19
TAH	39	Actual productivity rarely equals potential productivity	B	KY	08

		because of losses from communication and motivation problems		TAH TAH	16 22
TAH	40	The model accurately portrays management's inclination not to adjust the project's scheduled completion date during most of the development phase. Adjustments are instead made in the project's work force level. This behaviour pattern arises, according to DeMarco, for political reasons	B	JS	RC15
TAH	41	As the date approached, pressures developed that overrode the work force stability: management became increasingly willing to "pay any price" necessary to avoid overshooting the "Maximum Tolerable Completion Date"	B	JS NB	RC15 05
Model Behaviour					
TAH	42	Using the system dynamics model as an experimentation vehicle can provide many benefits to the software engineering community	A		
TAH	43	In addition to permitting less costly and less time consuming experimentation, simulation models make perfectly controlled experiments possible	A		
TAH	44	By using a model of a complex system, more can be learned about internal interactions than would ever be possible through manipulation of the real system	A		
TAH	45	A full-time employee allocates, on the average, 60% of his or her time to productive work on the project	B	JS	RC15
TAH	46	Most software projects start with a smaller core of designers, and as the project develops, the work force slowly builds up to higher levels	B	JS	RC15
TAH	47	In the early stages of a project, project managers are generally willing to adjust the work force level to maintain the project on its scheduled course. However, as the project proceeds, management becomes increasingly reluctant to add new people out of an increasing desire that the work force stabilise	B	NB JS	05 RC15
TAH	48	The shift away from work force adjustments to schedule adjustments continues as the project progresses	B	JS	RC15
TAH	49	Project members are not willing to maintain an above-normal work rate indefinitely. Once people start working at a rate above their normal rate, their "Overwork Duration Threshold" decreases because people enjoy and need their slack time	B	JS	RC15
On the accuracy of software estimation					
TAH	50	For many years, estimation of software project development time and cost had been an intuitive process based on experience and analogy {200}	B	JS	RC15
TAH	51	Having captured within our integrative system dynamics model "influence variables of software development and their causal relationships", we embark on a quantitative analysis of software cost and schedule estimation	B	TAH	43
TAH	52	By imposing different estimates on a software project we create different projects. Research clearly indicates that the decisions that people make in projects and the actions they take are significantly influenced by the pressures and perceptions produced by the project's schedule {57,109,125,224,236}	A	JS	RC15

TAH	53	A higher work force level generally means more communication and training overhead, which in turn affect productivity negatively. Scheduling can dramatically change the manpower loading throughout the life of a project	B	NB JS TAH	04 RC15 64
TAH	54	When the project is perceived to he ahead of schedule, "Parkinson's Law indicates that people will use the extra time for ... personal activities, catching up on the mail, etc. " {49}. This, of course, means that they become less productive	A	JS	RC15
TAH	55	The management reserves ranged from 5% to 50% of the estimated software cost with a mean of 18% {85}	B	JS	RC15
TAH	56	The project's initial estimates create pressures and perceptions that affect how people behave on the project. Overestimating a project often leads to an expansion of the project members' slack time activities, which leads to further reductions in productivity	B	JS	RC15
TAH	57	The "Safety Factor Policy" does achieve its intended objective: more accurate estimates. However, the organisation pays dearly for this	A	JS	RC15
Portability of estimation models					
TAH	58	Portability of the models has proven to be especially poor {43,49,144,184}. In the case of a hypothetical software project of 36,000 machine-language executable instructions the highest (of 12) estimate is over 650% higher than the lowest	A	JS	RC15
TAH	59	A software cost estimation model should avoid the use of variables whose values cannot be determined until the project is complete	B	JS	RC15
TAH	60	The policy of allocating project members half-time to the project results in a cost that is about 22% higher	B	JS	RC15
TAH	61	Different policies affect what the project's cost will end up being and should therefore be explicitly considered when project cost estimates are made	A	JS	RC15
TAH	62	More people on the project means more work gets done. It also means that the project team's overall productivity is lower because of the increased communication and training overheads	B	JS TAH	RC15 53
TAH	63	Variables used in cost estimation tend to be those which are easier to measure, quantify, and estimate, even if they are not the most significant {65}	B	JS	RC15
TAH	64	Our model results indicate "adding manpower to a late software project makes it more costly, but *not necessarily later*"	A	TAH AvD	06 Remark 02
TAH	65	A different distribution of estimated effort among a project's phases creates a different project	A	JS	RC15
Analogy method of software estimation					
TAH	66	A project's estimate creates pressures and perceptions that directly influence the decisions people make and the actions they take throughout the project's life cycle	B	JS TAH	RC15 65
TAH	67	Because of the inherent tendency to overshoot, the use of the analogy method in estimating injects a bias in scheduling, a bias that in the long-run generates longer than necessary schedules. The phenomenon of projects consuming longer and longer schedules is one that has been frequently encountered in system dynamics studies of organisational behaviour {241}	A	JS	RC15

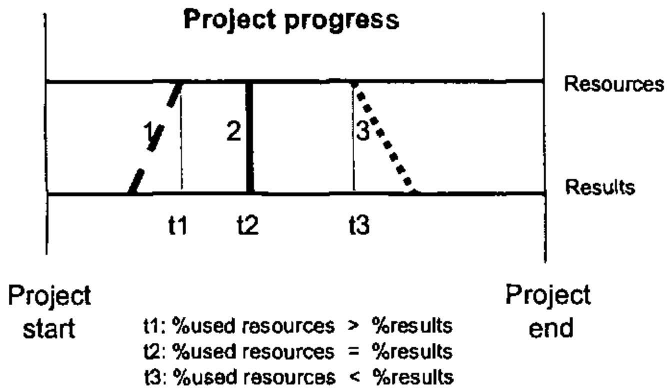
TAH	68	When a software development project overruns its schedule there is an apparent cause: the project was poorly estimated	B	JS	RC15
TAH	69	After all, software estimation is not yet an exact science. Significantly, it is often impossible in a real life situation to demonstrate that underestimation was *not* in fact the cause	B	JS	RC15
The 90% syndrome					
TAH	70	Estimates of the fraction of work completed increase as originally planned until a level of about 80-90% is reached. The programmer's individual estimates then increase only very slowly until the task is actually completed" {28}	A	JS	RC15
TAH	71	The "90% syndrome" arises because of the interaction of two factors: underestimation and imprecise measurement of progress	B	JS JS TAH	RC05 RC15 11
TAH	72	The better the measurement tool the earlier it will detect that progress is not keeping up with the underestimated schedule	A	JS TAH	RC15 11
TAH	73	The result of sticking with a schedule that is too tight is often an increase in the project's cost {49} due to a large work force level	B	JS	RC15
The economics of quality assurance					
TAH	74	A significant feature of the relationship between the QA effort expended and the percentage of errors detected during development, is the diminishing returns as QA expenditures exceed 20-30% of development effort	B	JS	RC15
TAH	75	QA is used not only in the development phase but also to minimise the cost of the testing phase	B	JS	RC15
TAH	76	QA policy does have a significant impact on total project cost	A	JS	RC15
Some conclusions					
TAH	77	An integrated approach helps us achieve an overall understanding	A		
TAH	78	The model identifies feedback mechanisms and uses them to structure and clarify relationships in software project management	A		
TAH	79	The schedule overshoot problem can arise not only because of schedule underestimation, but also because of management's hiring policies	A		
TAH	80	A different schedule creates a different project	A		
TAH	81	An important implication is that a software estimation model cannot be judged solely based on how accurately it estimates historical projects	A		
TAH	82	Evidence in the literature indicates that currently available quantitative software estimation tools are not particularly portable from the company in which they were developed to another (e.g., see {43,49}	A		

*) {...} = reference number in the list of references included in the book [Abdel-Hamid & Madnick 1991]
A = → Table 3.8.2
B = → Table 3.8.3

Table 3.8.1: Success / failure factors "Software Project Dynamics – An Integrated Approach"

3.8.3 Remarks

Au-thor	No	Re-mark	Description
TAH	05	01	The behaviour of an individual subsystem in isolation may be different from its behaviour when it interacts with other subsystems {68} *Remarks (AvD):* *- Within an information system this may be set off by means of carrying out an integration test. It does often happen that, as a result of a breakdown in communication, the sub systems do not quite attune to each other. This is discovered during the integration test and as a result, action is taken.* *- It is also possible that problems between information systems occur. A tested information system that has undergone a successful acceptance test by its users may for example start behaving like a "Trojan Horse" once it has become part of the production environment, through unjustly using resources for longer periods of time. This does occur relatively often.*
TAH	06	02	Brooks' Law: adding more people to a late software project makes it later {57}.
TAH	64		Our model results indicate "adding manpower to a late software project makes it more costly, but *not necessarily later*" *Remark (AvD):* *I do agree with TAH 64, although in several situations such as the CCIP and Telephony projects, I opted for not bringing extra manpower into action in the last phase of the project. Instead, a few colleagues and I worked many extra hours over a period of several weeks. However, that does not alter the fact that in cases where it is possible to use extra staff it can be useful in the last phase of a project when those employees did know the environment well or have for example specific technical knowledge which can be put to good use immediately. The latter does usually involve high costs because in those cases it may be necessary to fly in staff from another part of the world.*
TAH	09	03	System complexity grows as the square of the number of system elements; therefore, experience with a small system cannot account for all the things that will have to be done in a large system. Neither will the Experience Method apply to systems of totally different content {23}
TAH	14		The relationship between cost and system size is not linear. In fact, cost increases approximately exponentially as size increases {188} *Remark (AvD):* *Only few project leaders and software engineers are aware of the fact that the complexity and the costs of an information system are not linear but are in fact quadratic (complexity) respectively exponential (costs) in relation to the size of the information system. This was also proven in the Multihouse versus Nutsbedrijven case (paragraph 5.11.1).*
TAH	10	04	In a multi-project environment, competition for company resources becomes a significant dimension *Remark (AvD):* *As a project leader, you are often confronted by this. Employees are given an extra task by their line manager, however the other jobs also still need to continue. Furthermore, the mutual priority is not indicated. This makes things very difficult for the employees in question. Several project leaders are tugging away at them but nobody is satisfied with their achievements. This is very awkward and leads to less productivity and reduced motivation (when*

			everything has priority, nothing has priority anymore). Things like this also played a part in the ACCINT business case (paragraph 5.11.3).
TAH	11	05	It is difficult to measure performance in programming {181} --- *Remark (AvD):* *TAH/34: Progress, especially in the earlier phases of software development, is measured by the rate of expenditure of resources rather than by some count of accomplishments {81}.* *This often occurs but can lead to disappointment in the further course of the project. Both the used resources as well as the progress (results) need to be measured. The figure below shows that the used resources and results are in balance at a vertical line. This is the case with line 2 at point in time t2. In that case, for example, 35% of the available resources have been used and 35% of the required results have been achieved. Line 1 at point in time t1 indicates that the results are running behind with the used resources at point in time t1. Line 3 at point in time t3 shows that the results are ahead of the used resources.*  *Figure 3.8.1: Project progress*

"It is difficult to measure performance in programming {181}". This is often the case, although there are of course possibilities for measuring the progress. For instance, the OKAPI project (business case in section 5.6). The online part of OKAPI consists of 132 display images that are divided over five sub systems. The figure below is a (schematic) example of a display image. The coherence of the 132 display images was shown during the construction in so-called display image communication diagrams. The large drawings were hanging on the wall. The four corners of a display image were used for showing the progress:

- *corner1 was coloured green as soon as the program in question (dialogue stated) was coded;*
- *corner 2 was coloured blue as soon as the dialogue in question was tested;*
- *corner 3 was coloured yellow as soon as the dialogue in question was tested in coherence with the surrounding dialogues (integration test);*
- *corner 4 was coloured red as soon as the acceptance test by the key user in question had taken place successfully.*

That way any interested employee/ manager was able to follow the progress by looking at the wall of the project room.
Providing the dialogues with a weighting factor in advance made it possible to establish the percentage of the progress using a simple calculation programme. It was important that an evaluation with regard to the weighting factor took place after realisation of a dialogue. Was this correct? If not, then what were the consequences for other dialogues yet to be realised? Changes to the weighting factor of yet to be realised dialogues sometimes did have consequences for the planning and progress. Over the course of the project, the changes with regard to weighting factors became less through advancing insight.
The wall overview with or without (partly) coloured display images did present a clear idea of the progress of the project and was highly appreciated by the management.

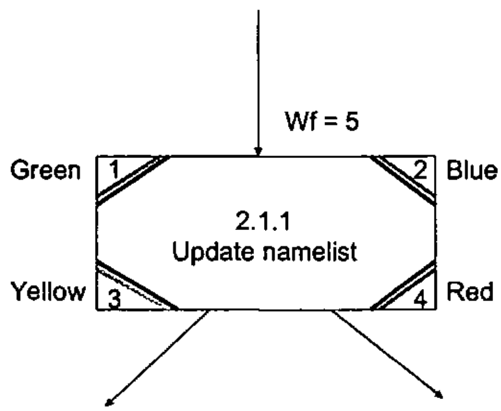


Figure 3.8.2: Measuring the progress of the OKAPI project

3.8.4 SUFFI tables TAH A and B

Au-thor	No	Description
Introduction		
TAH	09	System complexity grows as the square of the number of system elements; therefore, experience with a small system cannot account for all the things that will have to be done in a large system. Neither will the Experience Method apply to systems of totally different content {23} *)
TAH	12	The longer an error goes undetected, the more extensive the necessary rework and the greater the cost
TAH	14	The relationship between cost and system size is not linear. In fact, cost increases approximately exponentially as size increases {188}
TAH	16	A major challenge to managers is to motivate employees. A mechanism for motivation, which is attracting interest in the software engineering field, is "goal setting" {49}
Human Resource Management		
TAH	18	Newly hired project members pass through an orientation during which they are less than fully productive {75,243}
Software production / development		
TAH	25	When a project is perceived to be behind schedule, people tend to work harder to bring it back on schedule. There is a threshold beyond which employees would not be willing to work at an "above-normal" rate {169}
TAH	30	Schedule pressures often result in the "overlapping of activities that would have been accomplished better sequentially", and overlapping can significantly increase the chance of errors {252}
System testing		
TAH	31	We assume that undetected errors will become either "Active errors" (errors that produces more errors) or "Passive errors". Because design specs are the blueprints of the system's code, any errors in design will get translated into coding errors
Model Behaviour		
TAH	42	Using the system dynamics model as an experimentation vehicle can provide many benefits to the software engineering community
TAH	43	In addition to permitting less costly and less time consuming experimentation, simulation models make perfectly controlled experiments possible
TAH	44	By using a model of a complex system, more can be learned about internal interactions than would ever be possible through manipulation of the real system
On the accuracy of software estimation		
TAH	52	By imposing different estimates on a software project we create different projects. Research clearly indicates that the decisions that people make in projects and the actions they take are significantly influenced by the pressures and perceptions produced by the project's schedule {57,109,125,224,236}
TAH	54	When the project is perceived to be ahead of schedule, "Parkinson's Law indicates that people will use the extra time for … personal activities, catching up on the mail, etc. " {49}. This, of course, means that they become less productive
TAH	57	The "Safety Factor Policy" does achieve its intended objective: more accurate estimates. However, the organisation pays dearly for this
Portability of estimation models		
TAH	58	Portability of the models has proven to be especially poor {43,49,144,184}. In the case of a hypothetical software project of 36,000 machine-language executable instructions the highest (of 12) estimate is over 650% higher than the lowest
TAH	61	Different policies affect what the project's cost will end up being and should therefore be explicitly considered when project cost estimates are made
TAH	64	Our model results indicate "adding manpower to a late software project makes it more costly, but *not necessarily later*"

TAH	65	A different distribution of estimated effort among a project's phases creates a different project
Analogy method of software estimation		
TAH	67	Because of the inherent tendency to overshoot, the use of the analogy method in estimating injects a bias in scheduling, a bias that in the long-run generates longer than necessary schedules. The phenomenon of projects consuming longer and longer schedules is one that has been frequently encountered in system dynamics studies of organisational behaviour {241}
The 90% syndrome		
TAH	70	Estimates of the fraction of work completed increase as originally planned until a level of about 80-90% is reached. The programmer's individual estimates then increase only very slowly until the task is actually completed" {28}
TAH	72	The better the measurement tool the earlier it will detect that progress is not keeping up with the underestimated schedule
The economics of quality assurance		
TAH	76	QA policy does have a significant impact on total project cost
Some conclusions		
TAH	77	An integrated approach helps us achieve an overall understanding
TAH	78	The model identifies feedback mechanisms and uses them to structure and clarify relationships in software project management
TAH	79	The schedule overshoot problem can arise not only because of schedule underestimation, but also because of management's hiring policies
TAH	80	A different schedule creates a different project
TAH	81	An important implication is that a software estimation model cannot be judged solely based on how accurately it estimates historical projects
TAH	82	Evidence in the literature indicates that currently available quantitative software estimation tools are not particularly portable from the company in which they were developed to another (e.g., see {43,49}

*) {...} = reference number in the list of references included in the book
[Abdel-Hamid & Madnick 1991]

Table 3.8.2: Success / failure factors "Software Project Dynamics – An Integrated Approach" Category A

Au-thor	No	Description
Introduction		
TAH	01	Software engineering encompasses both the technical aspects of software development as well as the managerial ones {33,47,246} *)
TAH	02	Poor management can increase software costs more rapidly than any other factor {261}
TAH	03	We still lack the fundamental understanding of the software development process {87,96,173} and without such an understanding the likelihood of any significant gains in the management of software development front is questionable {36,130,161}
TAH	04	There are hundreds of variables that affect software development. Furthermore, these variables are not independent; many of them are related to one another {119,189}. THA constructed *a holistic model* of the software development process
TAH	05	The behaviour of an individual subsystem in isolation may be different from its behaviour when it interacts with other subsystems {68}
TAH	06	Brooks' Law: adding more people to a late software project makes it later {57}
TAH	07	People under time pressure don't work better, they just work faster. ... In the struggle to deliver any software at all, the first casualty has been consideration of the quality of the software delivered {81}

TAH	08	Two factors significantly influence the initial estimate of the job size: (1) the firm's previous experience; and (2) the general over-all tendency to underestimate the job size {49,223}
TAH	10	In a multi-project environment, competition for company resources becomes a significant dimension
TAH	11	It is difficult to measure performance in programming {181}
TAH	13	Inadequate planning is the primary reason for loss of control on many computer programming projects {208}
TAH	15	Frequent budget manipulations by management to avoid overruns make historical cost data questionable {188}
TAH	17	All programmers are optimists. They always assume that "This time it will surely run" or "I just found the last bug" {57}
Human Resource Management		
TAH	19	The training of newcomers, both technical and social, is usually carried out by the "old-timers". This is costly because "while the old-timer is helping the new employee learn the job, his own productivity on his other work is reduced {55,74,77}
TAH	20	Toward the end of the project there is likely to be reluctance to bring in new people
Software production / development		
TAH	21	Under schedule pressures, walk throughs and inspections are usually the greatest casualties, they are not only relaxed but often suspended altogether {94}
TAH	22	Problems of communication and motivation are responsible for inadequacies in process and for consequent losses in productivity {240}
TAH	23	Work force experience level and increases in project familiarity due to learning-curve effects affect the productivity of a software development project {76,234,261}
TAH	24	The project goals and schedules can play a significant motivational role throughout the life of a software development project
TAH	26	When project members perceive some "excesses" in the schedule (the case of negative schedule pressure), some excesses will be "absorbed" by the workers as "under-work" before downward adjustments are made in the project's schedule {49,131}. As with positive schedule pressures, there are limits on how in the case of negative schedule pressure employees are willing or allowed to absorb
TAH	27	Communication overhead increases in proportion to n^2, where n is the size of the team {57,180,231,236,273}
TAH	28	Design errors in the early design phase are generated at a higher rate than are coding errors {165}
TAH	29	Newly hired employees are not only less productive on average but also more error-prone than their experienced counterparts {92,189}
System testing		
TAH	32	The earlier the undetected error is, the more "generations" of errors it will produce, and thus the more costly it will end up being
Controlling / planning		
TAH	33	Progress in a software project is measured by the number of resources consumed, tasks completed, or both
TAH	34	Progress, especially in the earlier phases of software development, is measured by the rate of expenditure of resources rather than by some count of accomplishments {81}
TAH	35	As a software project develops, project members often realise that they have under-estimated the number of tasks that constitutes the software system being developed {61}
TAH	36	There is a powerful tendency to focus on the highly visible mainline components of the software, and to underestimate or completely miss the unobtrusive components (e.g., help message processing, error processing, and moving data around {49}

TAH	37	In some cases the management becomes increasingly willing to "pay any price" necessary to avoid overshooting the "Maximum Tolerable Completion Date". In such cases, management is often willing to hire more people
Case		
TAH	38	At the System Development Section 25% of an experienced employee's time is committed per new employee
TAH	39	Actual productivity rarely equals potential productivity because of losses from communication and motivation problems
TAH	40	The model accurately portrays management's inclination not to adjust the project's scheduled completion date during most of the development phase. Adjustments are instead made in the project's work force level. This behaviour pattern arises, according to DeMarco, for political reasons
TAH	41	As the date approached, pressures developed that overrode the work force stability: management became increasingly willing to "pay any price" necessary to avoid overshooting the "Maximum Tolerable Completion Date"
Model Behaviour		
TAH	45	A full-time employee allocates, on the average, 60% of his or her time to productive work on the project
TAH	46	Most software projects start with a smaller core of designers, and as the project develops, the work force slowly builds up to higher levels
TAH	47	In the early stages of a project, project managers are generally willing to adjust the work force level to maintain the project on its scheduled course. However, as the project proceeds, management becomes increasingly reluctant to add new people out of an increasing desire that the work force stabilise
TAH	48	The shift away from work force adjustments to schedule adjustments continues as the project progresses
TAH	49	Project members are not willing to maintain an above-normal work rate indefinitely. Once people start working at a rate above their normal rate, their "Overwork Duration Threshold" decreases because people enjoy and need their slack time
On the accuracy of software estimation		
TAH	50	For many years, estimation of software project development time and cost had been an intuitive process based on experience and analogy {200}
TAH	51	Having captured within our integrative system dynamics model "influence variables of software development and their causal relationships", we embark on a quantitative analysis of software cost and schedule estimation
TAH	53	A higher work force level generally means more communication and training overhead, which in turn affect productivity negatively. Scheduling can dramatically change the manpower loading throughout the life of a project
TAH	55	The management reserves ranged from 5% to 50% of the estimated software cost with a mean of 18% {85}
TAH	56	The project's initial estimates create pressures and perceptions that affect how people behave on the project. Overestimating a project often leads to an expansion of the project members' slack time activities, which leads to further reductions in productivity
Portability of estimation models		
TAH	59	A software cost estimation model should avoid the use of variables whose values cannot be determined until the project is complete
TAH	60	The policy of allocating project members half-time to the project results in a cost that is about 22% higher
TAH	62	More people on the project means more work gets done. It also means that the project team's overall productivity is lower because of the increased communication and training overheads
TAH	63	Variables used in cost estimation tend to be those which are easier to measure, quantify, and estimate, even if they are not the most significant {65}
Analogy method of software estimation		

TAH	66	A project's estimate creates pressures and perceptions that directly influence the decisions people make and the actions they take throughout the project's life cycle
TAH	68	When a software development project overruns its schedule there is an apparent cause: the project was poorly estimated
TAH	69	After all, software estimation is not yet an exact science. Significantly, it is often impossible in a real life situation to demonstrate that underestimation was *not* in fact the cause
The 90% syndrome		
TAH	71	The "90% syndrome" arises because of the interaction of two factors: underestimation and imprecise measurement of progress
TAH	73	The result of sticking with a schedule that is too tight is often an increase in the project's cost {49} due to a large work force level
The economics of quality assurance		
TAH	74	A significant feature of the relationship between the QA effort expended and the percentage of errors detected during development, is the diminishing returns as QA expenditures exceed 20-30% of development effort
TAH	75	QA is used not only in the development phase but also to minimise the cost of the testing phase

*) {...} = reference number in the list of references included in the book [Abdel-Hamid & Madnick 1991]

Table 3.8.3: Success / failure factors "Software Project Dynamics – An Integrated Approach" Category B

3.8.5 Conclusions

Additional

Some TAH SUFFIs are additional to the tables 3.7.11 (John Smith) and 3.7.12 (others). A few examples being:

TAH/09

System complexity grows as the square of the number of system elements; therefore, experience with a small system cannot account for all the things that will have to be done in a large system. Neither will the Experience Method apply to systems of totally different content {23}.

TAH/14

The relationship between cost and system size is not linear. In fact, cost increases approximately exponentially as size increases {188}.

TAH/52

By imposing different estimates on a software project we create different projects.

TAH/64

Our model results indicate "adding manpower to a late software project makes it more costly, but *not necessarily later.*

TAH/65

A different distribution of estimated effort among a project's phases creates a different project.

TAH/80

A different schedule creates a different project.

More insight

Not all TAH SUFFIs are "new SUFFIs" but they do mainly provide more insight and are more specific. "Poor Management" is for example Big Hitter 01 and rather generic. However, behind "Poor Management" a whole range of specific subjects are hidden that may contribute to this Big Hitter. A number of TAH SUFFIs do provide these subjects. As shown in the POTVIS project (par. 5.2.3), some TAH SUFFIs can influence each other *(TAH/52,TAH/54,TAH/57)*

For a number of the (successfully) realised projects the following applies:
If the management had been aware of the applicable TAH SUFFIs, then the project would have run smoother and the costs would have been lower".

For the audited projects goes:
"If the management had been aware of the applicable TAH SUFFIs, then the results of the project would have been better".

Organisation

In the case of failing project management, one has to remember that a large part of the errors are not incidental but are caused by the working method in an organisation and thus come under the responsibility of the management [Rijsenbrij & Kouwenhoven 1996, audit ACCINT].

TAH SUFFIs

Based upon the above mentioned, it can be decided that the collection of TAH SUFFIs is of major importance to starting as well as experienced project leaders as well as the management of organisations. Studying of these SUFFIs in advance, can prevent many problems regarding ICT projects.

3.8.6 The SUFFI model

Tables 3.7.11 (John Smith), 3.7.12 (others) and 3.8.2 (Abdel-Hamid) include the results with regard to the found SUFFIs. Together these form the SUFFI model. This is shown in figure 3.8.1. In next chapter the SUFFI model will be extended with "big hitters" (table 4.2).

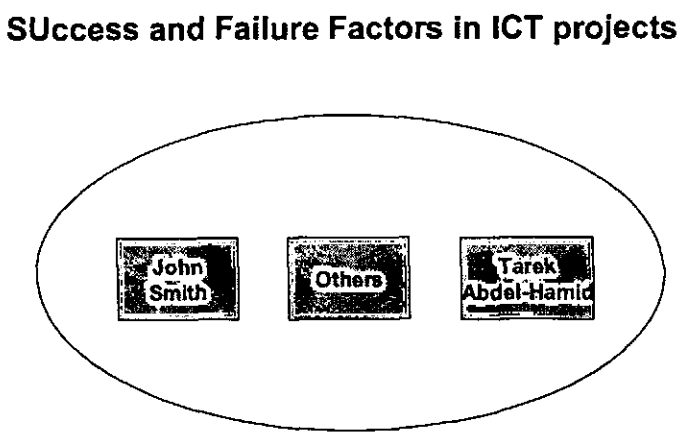


Figure 3.8.1: SUFFI model, consists of the SUFFIs from John Smith, Tarek Abdel-Hamid and others

CHAPTER 4

WHICH ROOT CAUSES ARE THE 'BIG HITTERS'?

4.1 VIEWS OF THE AUTHORS

This chapter lists the views of the authors on the most important (most common) success and failure factors, followed by a conclusion.

John Smith

Any of the root causes can have a profound impact on the outcome of a project, and the problem is that they hunt in packs. Many troubled projects are beset by, perhaps, half a dozen of these root causes. It is *unwise* to try to rank the root causes into 'big hitters' and 'the rest'. However, such ranking is definitely of value for an *individual* project as it will inform the prioritisation of 'turnaround' actions. [Smith 2001, Smith 2002]

Capers Jones

- capable managers and capable technical personnel are needed;
- attempting to construct large software projects without adequate management and quality control tools is not a safe undertaking;
- not only are large systems expensive, but they also have one of the highest failure rates of any manufactured object in human history;
- in terms of their ability to build large software applications successfully, the rank order of six sub industries is: systems software, outsource vendors, commercial software, military software, management information software, end-user software.

[Jones 1996-1]

Michael Evans et al.

The *seven common characteristics* are:

1. failure to Apply Essential Project Management Practices (57%);
2. unwarranted Optimism and Unrealistic Management Expectations (41%);
3. failure to Implement Effective Software Processes (30%);
4. premature Declarations of Victory (20%);
5. lack of Program Management Leadership (13%);
6. untimely Decision-Making (8%);
7. lack of Proactive Risk Management (3%).

Michael Evans: "The ranking of the characteristics is by frequency of characteristic occurrence; therefore, the data show what may be the likely dysfunctional causes, *but not their relative impact on projects or programs.*"

[Evans et al. 2002]

Watts Humphrey: "*Poor project management* will defeat good engineering, and is the most frequent cause of project failure."

[Evans et al. 2002, Humphrey 1998]

K.T. Yeo

Table 4.1 lists the *top five factors* categorised under Sp, S1 and S2 sphere of influence respectively.

Rank	Sp Process driven issues	S1 Context driven issues	S2 Content driven issues
1	Underestimate of timeline	Lack user involvement and inputs from the onset	Consultant/vendor underestimated the project scope and complexity
2	Weak definitions of requirements and scope	Top down management style	Incomplete specifications when project started
3	Inadequate project risk analysis	Poor internal communication	Inappropriate choice of software
4	Incorrect assumptions regarding risk analysis	Absence of an influential champion and change agent	Changes in design specifications late the project
5	Ambiguous business needs and unclear vision	Reactive and not pro-active in dealing with problems	Involve high degree of customisation in application

Table 4.1: Top 5 failure factors under Sp, S1 and S2 [Yeo 2002]

Lorin May

The failure causes are:

1. poor user input;
2. stakeholder conflicts;
3. vague requirements;
4. poor cost and schedule estimation;
5. skills that do not match the job;
6. hidden costs of going 'Lean and Mean';
7. failure to plan;
8. communication breakdowns;
9. poor architecture;
10. late failure warning signals.

Lorin May: "The factors of successful project management have been documented for years – they merely need greater attention."
[May 1998]

John S. Reel

The *five essential factors* to managing a successful software project are:

1. start on the right foot;
2. maintain momentum;
3. track progress;
4. make smart decisions;
5. institutionalise post-mortem analyses.

If you master these five critical success factors, you greatly increase the odds of completing your project on time and within budget. Just as important, you increase your chances of actually delivering something your users want.

Start on the right foot

Just as it is difficult to grow strong plants in weak soil, it is almost impossible to successfully lead a development effort that is set up improperly. Tom Field [1997] analysed pitfalls in software development efforts and gave 10 signs of IS project failures:

1. project managers don't understand users' needs;
2. the project's scope is ill-defined;
3. project changes are managed poorly;
4. the chosen technology changes;
5. business needs change;
6. deadlines are unrealistic;
7. users are resistant;
8. sponsorship is lost;
9. the project lacks people with appropriate skills;
10. managers ignore best practices and lessons learned.

[Reel 1999]

Jan Oonincx

The book discusses, grouped in seventeen chapters, a number of aspects that often cause development and introduction of information systems to fail:

- information systems, which are set up too ambitious;
- information systems, which are set up too isolated;
- information systems, which are set up without proper planning;
- insufficient involvement of future users in the development of information systems;
- a passive attitude of the top management also often leads to disappointing results.

[Oonincx 1982]

Nico Beenker

The *five most often mentioned factors*, starting with the factor that is mentioned most often, are:

1. planning is too optimistic;
2. badly phrased contracts;
3. poor project management;
4. poor communication;
5. problems escalated too late for intervention.

[Beenker 2004]

Peter Noordam et al.

In a KPMG study [2004] amongst 252 controllers, the respondents named the following reasons as being the most important ones for the failure of projects:

1. poor project management (32 percent);
2. lack of communication in and around the project (20 percent);
3. objectives not defined (17 percent);
4. unfamiliarity with scope and complexity (17 percent)
5. technical complexity and technical integration issues (7 percent);
6. others (unknown) (7 percent).

[Noordam et al. 2007, KPMG 2004]

This chapter lists the views of the authors on the most important (most common) success and failure factors. Unfortunately they use different terms:

- 'big hitters' [Smith 2001];
- seven common characteristics [Evans et al. 2002];
- the most frequent cause of project failure [Humphrey 1998];

- top 5 failure factors [Yeo 2002];
- the failure causes [May 1998];
- the five essential factors [Reel 1999];
- signs of IS project failures [Field 1997];
- aspects that often cause development and introduction of information systems to fail [Oonincx 1982];
- the most often mentioned factors [Beenker 2004];
- reasons as being the most important ones for the failure of projects [Noordam et al. 2007].

In order to be able to use one single phrase within this thesis, the term 'big hitter' (BH) by John Smith was selected.

4.2 SUMMARY

John Smith: "It is *unwise* to try to rank the root causes into 'big hitters' and 'the rest'. However, such ranking is definitely of value for an *individual* project as it will inform the prioritisation of 'turnaround' actions". With regard to this particular point, John Smith disagrees with other researchers (Evans, Yeo, Reel, Beenker, Noordam) that do indicate a ranking of success/failure factors. Table 4.2 includes the success/failure factors that have been named as "big hitter" by at least 4 authors.
Together with tables 3.7.11, 3.7.12 and 3.8.2, table 4.2 forms the ***"Reference model success and failure factors ICT projects"*** *(for short: **SUFFI model**, SUFFI = SUccess/Failure Factors in ICT projects) (see figure 4.1).* That establishes: "The opinion of others about Software Project Management".

Success/failure factors that are mentioned the "big hitters"(BH)	**CJ**	**ME**	**KY**	**LM**	**JR**	**JO**	**NB**	**PN**	**Total**
Poor project management (BH01)	+	+	+	+	+	+	+	+	8
Deadlines are unrealistic (BH02)		+	+		+		+		4
Poor communication (BH03)			+	+			+	+	4
Incomplete/weak definition requirements (BH04)			+	+	+			+	4
Insufficient involvement of future users (BH05)			+	+	+	+			4
John Smith: "It is *unwise* to try to rank the root causes into 'big hitters' and 'the rest'. However, such ranking is definitely of value for an *individual* project as it will inform the prioritisation of 'turnaround' actions."									

Table 4.2: Success/failure factors that are mentioned the "big hitters"

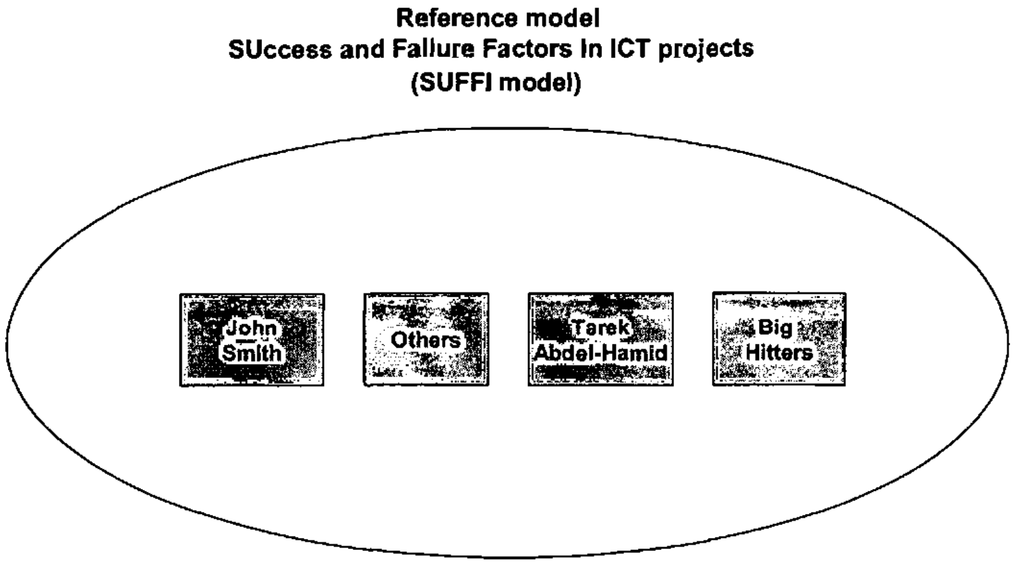


Figure 4.1: Reference model SUccess and Failure Factors in ICT projects (SUFFI model)

4.3 DEFINITION OF THE PROBLEM [Answers to sub questions 2 and 3]

Definition of the problem, sub question 2:

Is it possible to derive SUFFIs from international publications, Dutch publications and from the procedures in Tarek Abdel-Hamid's work on Software Project Management and if so, what particular SUFFIs?

Answer:

Together with tables 3.7.11, 3.7.12 and 3.8.2, table 4.2 forms the *"Reference model success and failure factors ICT projects" (for short: SUFFI model, SUFFI = SUccess/Failure Factors in ICT projects) (see figure 4.1).* That establishes "The opinion of others about Software Project Management".

Definition of the problem, sub question 3:

Are there any Big Hitters amongst the SUFFIs and if so what are these?

Answer:

Table 4.2 includes the success/failure factors that have been named as "big hitter".

CHAPTER 5

THE PORTFOLIO OF PROJECTS

5.1 INTRODUCTION

In chapter 3 and chapter 4 *"The opinion of others about Software Project Management"* is mapped using the "Reference model success and failure factors in ICT projects (SUFFI model)".
In this chapter sub question 4 is at the centre: *"Which SUFFIs are applicable to what particular project from the portfolio of the author's (AvD) projects?"*.

The following will come up for discussion: "*a look at how the projects the author (AvD) worked on were managed and how they agree or disagree with what others say happens, not to apportion blame but as an academic exercise in finding out the true methods used in practice. The key here is the author's (AvD) observations and experience.*

Table 5.1 contains 9 projects and 4 project audits I completed. About these projects I wrote 12 project-based publications in Dutch journals. These 12 publications and 33 internal publications about these projects are in my possession. The 9 projects represent an effort and duration of about 16 years. My role in projects 6-9 was: *internal* project manager at the Delft University of Technology. Regarding the projects 1-5, I was the *external* project manager.
The 9 projects on which external publications in the trade magazines appeared are discussed in sections 5.2 up to and including 5.10. In the part of IT auditor, I audited a few "troubled projects". In section 5.11, four projects audits come up for discussion.

Remark in advance

Chapter 1, refers to research carried out by The American "Standish Group" [2003]. In their research, they aim even more emphatically at success and failure factors regarding ICT projects. My experiences with projects within ICT over the last 43 years divert from these research results. Not one of my projects was aborted prematurely. Some projects took more time/money than anticipated or were temporarily classed as "troubled project" but practically all projects were successfully implemented. Only the results of two projects I was involved with have not been put into use:

- the CCIP project for the FORTIS BANK Netherlands. This project did produce the required result: an application (SCOPUS) was procured by FORTIS and was adjusted for FORTIS to be used for the support of operational ITIL processes. In the week, that the project result was delivered it was announced that the supplier of the SCOPUS application had been bought up by SIEBEL. For the FORTIS management, this was reason to put the project result on ice till further notice;

- during my participation in a project for the Dutch Ministry of Defence there was a sudden order resulting in all external staff, through financial difficulties, having to leave the (sizeable) project at the end of the month in question. I was in the middle of my activities and therefore not able to conclude my part in the project properly.

In chapter 9: "DISCUSSION OF RESULTS", some aspects (Big Hitters) of some portfolio projects will come up for discussion.

No	Project with external project-based publications	Company / Remarks	Period	Duration in years	Ref. number in journal publications	Paragraph in this chapter
1	POTVIS	KLPD (National Police Services Agency) • *Improvement to infrastructure in a complete SAP-environment: network segregation, RBAC (Role-based Access Control), backup and restore, etc. (109 improvements).* • *Presentation of the results of RBAC at KLPD for the Dutch Society for Information Science NGI (80 participants).* **Staff** The project manager (AvD) was dedicated to the project. Among other things he designed the RBAC model. About fifteen staff members were part-time available for doing different jobs. **Size software** No application software development.	2003-2004	1.7	3	5.2
2	Kolibrie	KPN (Dutch Telecom company) • *Interfacing BAAN-ERP with other information systems (based on EAI (Enterprise Application Integration): message broker, adapters).* • *Presentation of the results of EAI at KPN for the Dutch Society for Information Architects (90 participants).* **Staff** The project manager (AvD) was dedicated to the sub project Interfacing. Among other things he designed the EAI architecture together with a BAAN-ERP interfacing specialist. Four other people participated in the sub project Interfaces. **Size software** 20,000 lines of code (estimated).	1999-2001	2	4 and 5	5.3

3	Charging method GAK	GAK/ASZ (Gemeenschappelijk Administratie Kantoor/Automatisering Sociale Zekerheid) • *Development of a charging method based on functional services (e.g. functional transaction codes).* • *Developed for the use with National databases (containing employees, employers and their contractual obligations).* **Staff** The project manager/research engineer (AvD) was dedicated to the research project. He developed the new charging method. About ten people gave input to the project. **Size software** No application software development.	1997-1998	0.5	6	5.4
4	IMPALA	DUT (Delft University of Technology) • *Implementation of a new PABX (Private Automatic Branch eXchange) with 7,000 telephone connections.* • *Charging modules, management procedures, information management, etc.* • *Project-based thesis Executive Master of IT Auditing (EMITA) EUR.* **Staff** The project manager (AvD) was dedicated to the sub project. Among other things he designed the IMPALA model. Two other staff members were available for doing different jobs. **Size software** 10,000 lines of code (estimated).	1993-1994	2	7	5.5

5	OKAPI	UoA (University of Amsterdam) • *Development of a new decentralised information system concerned with the financial building administration.* • *Including two-phase RBAC.* **Staff** The project manager (AvD) was dedicated to the project. Among other things he designed the functional design. Four other staff members were available for doing different jobs. **Size software** 90,000 lines of code (estimated).	1991-1994	2	8	5.6
6	GIRAF	DUT • *General Information Retrieval Facilities on mainframe computers.* • *Many applications on the DUT.* • *Flexible multi-language information system.* • *Adaptations possible without changing the system.* • *Sold to other Universities and companies in the Netherlands and Belgium.* • *After the project-based publication, some companies in the Netherlands & Belgium adapted the architectural/engineering solutions for all applications in their company.* **Staff** The project manager (AvD) was dedicated to the project. Among other things he designed the functional design and architecture of GIRAF. About three staff members were available for doing different jobs. **Size software** 160,000 lines of code.	1982-1984	2.5	9 and 10	5.7

7	AUBID	DUT • *Information system related to book reservations and book requests.* • *Based on CICS/VS (Customer Information Control System/Virtual Storage).* • *Some solutions, published in the project-based publication, are implemented by dozens companies (especially the solution of the printer problem).* **Staff** The project manager (AvD) was dedicated to the project. Among other things he designed the functional design. Four staff members were available part-time for doing different jobs. **Size software** 20,000 lines of code (estimated).	1977-1980	2.5	11 and 12	5.8
8	VDV	DUT • *Unique system to collect traffic data on behalf of the Research of Traffic Streams.* • *Very difficult to develop (hardware, software) because it was not clear in advance if a solution to the problem of the users (traffic engineers) could be found.* • *Engineering system for the traffic engineers of the DUT.* • *Has been used for more than 25 years.* **Staff** The project manager (AvD) was dedicated to the project. He designed and programmed the VDV model. Two staff members were available part-time for doing different jobs. **Size software** 6,000 lines of code.	1972-1974	3	13	5.9

9	BIBLIOSYSTEM	DUT • *Batch oriented information retrieval system.* • *Many applications, including bibliographies.* • *After the project-based publication, more than 100 companies asked for the documentation/sources.* **Staff** The project manager (AvD) was dedicated to the project. He designed and programmed the BIBLIOSYSTEM information system. One staff member was available part-time for doing different jobs. **Size software** 2,000 lines of code.	1970-1971	0.5	14	5.10
		Subtotal		*16.7*		
No	**IT-audits**	**Company / Remarks**	**Period**	**Duration in years**		
10	Multihouse	Multihouse BV • *Information system NUMIS-2000 developed by Multihouse BV.* • *Ordered by SWV NUMIS-2000.* • *Audit ordered by Court of Justice in Amsterdam.* • *Audit team (Commission of Third Party Experts): Prof.dr.ir. M. Looijen, Prof.dr.ir. G. C. Nielen and ir. A. J. van Dijk EMITA RE.* • *Both of the parties did not realise the degree of complexity.*	1997	0.2		5.11.1

11	SYSA	GOVERN • *Replacement of PROGA and PROGB by SAP R/3.* • *Although a lot of money had been invested in SAP/SYSA already, it was recommended to stop the project in order to avoid further escalation.* • *The head of ICT-Operations of the Information Provision Service had proposed to have a brief investigation (audit) done by an independent researcher/auditor.* • *The author (AvD) was asked to carry out this investigation. He accepted the commission under the express condition that he would be able to carry out an independent/impartial investigation.* • *It is interesting that six months on my advice with regard to PROGA was taken on board fully and an investigation into solution approach 1 with regard to PROGB as recommended by me was underway.*	2005	0.1		5.11.2
12	ACCINT	PUBLIC • *One of the objectives of the ACCINT project was: introduction of the Internet functionality (both the e-mail as well as the browse functionality) in a controlled and as safely as possible way for obvious users.* • *The "Rollout of the browse functionality" of the ACCINT project did not go entirely as planned.* • *The CIO (Chief Information Officer) of PUBLIC had commissioned the author (AvD) to carry out an audit/ investigation into ACCINT with the objective to provide an answer to the question "What went wrong?".* • *Although ICT-Operations had been receiving the necessary criticism in the report, the head of the department did agree with the conclusions.*	2005	0.1		5.11.3

13	SOX	FINANCE • *As part of SOX (the Sarbanes-Oxley Act), FINANCE, a well-known financial institution in the Netherlands, carried out audits/tests on the IT infrastructure.* • *The author (AvD) was one of the IT auditors and carried out audits on a number of realised IT projects.* • *SOX means: the adage "tell me" has changed into "show me". This has a major impact on the companies themselves, which was also obvious during these project audits.*	2006	0.4		5.11.4
		Subtotal		***0.8***		
		Total		***17.5***		

Table 5.1: Portfolio of projects

5.2 CASE: POTVIS (KLPD)

5.2.1 Introduction

From 1995 onwards, SAP R/3 was used by the Dutch National Police Agency (KLPD) for the division Operational Management. In 1995 and 1996, financial and logistics modules were brought into use. In 2003, HRM modules were added to these. Other applications, such as Employee Self Service (ESS), Customer Relation Management (CRM), Document/Record Management Systems (DMS), and Web Application Server (WAS), also form part of the picture. Summarising, one can say that SAP R/3 is a very important package for KLPD's business activities.

Audits

In 2001, the EDP AUDIT POOL (EAP) by order of the Audit Department of the Dutch Ministry of the Interior and Kingdom Relations carried out an audit.

The *Programma Verbetering Bedrijfsvoering (Programme Improvement Operational Management) (PVB)* of the KLPD has drawn up a list of recommendations, by means of the EAP's reports. In 2003, I was asked as an independent registered auditor (RE) to perform an audit into the status quo of the recommendations. The audit showed that a number of recommendations had not yet or not entirely been executed. The conclusion was that there were visible improvements but that a substantial amount of work still remained to be done. In connection with the results of the research and the importance of SAP R/3 to Operational Management, KLPD decided to start the POTVIS (PrOjecT Verbetering Infrastructuur SAP/Project Improvement Infrastructure SAP) project.

Objective POTVIS project

The objective of the POTVIS project (POTVIS=sperm whale) is: "*put in order the policy concerning the management and security of the SAP R/3 application and the SAP processing organisation by means of implementation of the not (entirely) realised recommendations. In doing so, the SAP R/3 application and the SAP R/3 environment is further professionalised resulting in adequate support of KLPD's Operational Management*".

Approach

The POTVIS project is realised (2003) in two phases. A number of crucial activities have been executed in phase 1 and the remaining activities will be carried out in phase 2.

I had been asked to act as project leader for the POTVIS project in phase 1.

One of the crucial activities that were executed in that phase 1 was "research into the functioning and improvement of the authorisation structure in relation with SAP R/3". This activity will be further discussed in the next paragraph.

5.2.2 Outline article RBAC (2004)

Introduction

Our society is changing, and the Dutch police are changing with it. This creates new challenges for supporting the police using modern ICT resources. This is why the 'Police ICT Plan' was generated. ICT support is also being further professionalized with respect to police enterprise activities. One of the packages that have been used within the Dutch National Police Agency (KLPD) for several years is SAP R/3. SAP R/3 *Enterprise* was deployed in July 2003. On this occasion, the KLPD fully revamped their authorisation concept. This represented a response to the needs of the KLPD and the control bodies, such as the audit service of the Ministry of Internal Affairs, to arrive at a modern, insightful and manageable authorisation concept. The new authorisation concept is based on 'role-based access control'. There is considerable (international) interest in this method.

This article presents a summary of the ICT authorisation policy of the Dutch police and provides insight into how the authorisation concept is implemented in the KLPD, and in particular in SAP R/3.

The article concludes with a summary of experience acquired to date.

Dutch Police Basic Security Level

The BBNP (*Basisbeveiligingsniveau Nederlandse Politie*) addresses basic security requirements and measures. The basic security measures are the measures employed for information security in each of the corps and the supra regional information systems.

The Police Information Security Regulations (*Regeling Informatiebeveiliging Politie*, or RIP for short) define information security as the *reliability of information provision*, which is regarded as a *component of quality assurance* for the enterprise processes and underlying information systems. The BBNP guideline is thus *aimed at the quality of the information provision process* and defines a minimum set of measures for ensuring the reliability (i.e. the availability, integrity and exclusivity) of the information systems.

Authorisation guide

Corps employees are granted specific authorisations to allow them to use information systems and the information in them. These authorisations are dependent on the *function* or *functional role* performed by the employee. An employee only receives access to information systems, and may only exercise authorisations with respect to the data stored in these systems, to the extent that this is necessary for his work ('*need to use*' and '*need to know*'). Before an employee can process data, certain things must be *arranged* on the user organisation side and certain things must be *configured* on the automation organisation side.

Role-Based Access Control (RBAC)

The concept of role-based access control (RBAC) plays an important role in access security. This is a methodical approach originating from the US National Institute of Standards and Technology (NIST). RBAC is based on *roles*, which are standardised sets of functions that are suitable for multiple users. Using functions and standardised sets of functions linked to them is not new [jp8], but a large amount of literature on the subject of RBAC has appeared in recent years.

In RBAC, the entities *users*, *subjects*, *objects*, *operations* and *permissions* play a prominent role, as do the relationships between these entities. *Users* are the entities that use an information system. A *subject* is a computer process or program that performs actions on behalf of a user. These actions are performed on an *object*, which is a resource that is accessible to the computer system. An *operation* is an action performed by a subject. *Permissions* (which are also called *privileges*) are authorisations to perform actions. Such an action involves a combination of an object and an operation.

Figure 5.2.1 shows the relationship between users, roles, permissions, operations and objects.

The 'subject' entity is not discussed in this article.

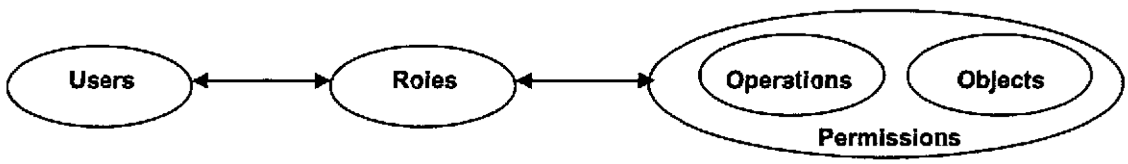


Figure 5.2.1: Relationship of RBAC components

In RBAC, it is thus not allowed to link permissions directly to users. Permissions are granted by means of two links: *a link between users and roles* and *a link between roles and permissions.* A significant advantage of this arrangement is that if a user is assigned a different position in the enterprise, only the user–role link has to be changed. This does not require a large amount of specialised knowledge. There are also major advantages with regard to disposition and control activities. As the number of roles is limited and mirrors the organisational structure, the number of changes will be small after an initial period. In addition to yielding considerably lower costs, using approved roles provides continual overview and insight into authorisation activities.

RBAC applications can be found at many different levels in ICT, such as in operating systems, database management systems, networks, workflow systems and Web services.

RBAC can also be used in Single Sign-On situations.

Authorisation in the KLPD

Based on the above, it can be concluded that:

1. *The 'need to know' and 'need to use' principles must form the basis for granting permissions or authorisations with respect to data, information and information systems.*
2. *Permissions or authorisations with respect to data, information and information systems must be granted using two links (see figure 5.2.1):*
   - *a link between users and functions (or functional roles);*
   - *a link between functions (or functional roles) and permissions (or authorisations).*
3. *Directly linking users to permissions or authorisations is **not** allowed.*

Authorisation in SAP R/3

Many software suppliers have responded to the international interest in role-based access control. RBAC can also be used in SAP R/3 Enterprise. SAP R/3 uses roles and can combine roles to form composite roles. In RBAC, the term 'role' is primarily used in the *functional sense.* In SAP R/3, the term 'role' refers to a *technical system* role. In the KLPD SAP R/3 authorisation concept, the terms *function* and *functional role* (which have a process-related nature) were chosen for the KLPD business view, while the terms *SAP R/3 role* and *SAP R/3 composite role* (which relate to technical aspects of the system) were chosen for the SAP R/3 system view.

The relationship between the business view and the system view is defined by a 1-to-1 link between a function or functional role and a SAP R/3 composite role (see figure 5.2.2).

Tasks are distinguished within a function or a functional role. SAP R/3 roles are clustered in a SAPR/3 composite role. The actual link is made inside SAP R/3 between the user ID and the SAP R/3 composite role. In this way, a user is linked based on his function or functional role (role-based).

The authorisation structure within RBAC and SAP R/3 is designated as 'object oriented'. An authorisation object forms the basis for determining the access privileges. It has at most ten fields for this purpose, which are defined in the data dictionary. The authorisation objects and associated authorisation fields and activity codes (values) are linked to a role via a profile (a set of authorisation objects). Besides authorisation objects, transactions can also be coupled to a SAP R/3 role via a menu (see figure 5.2.2).

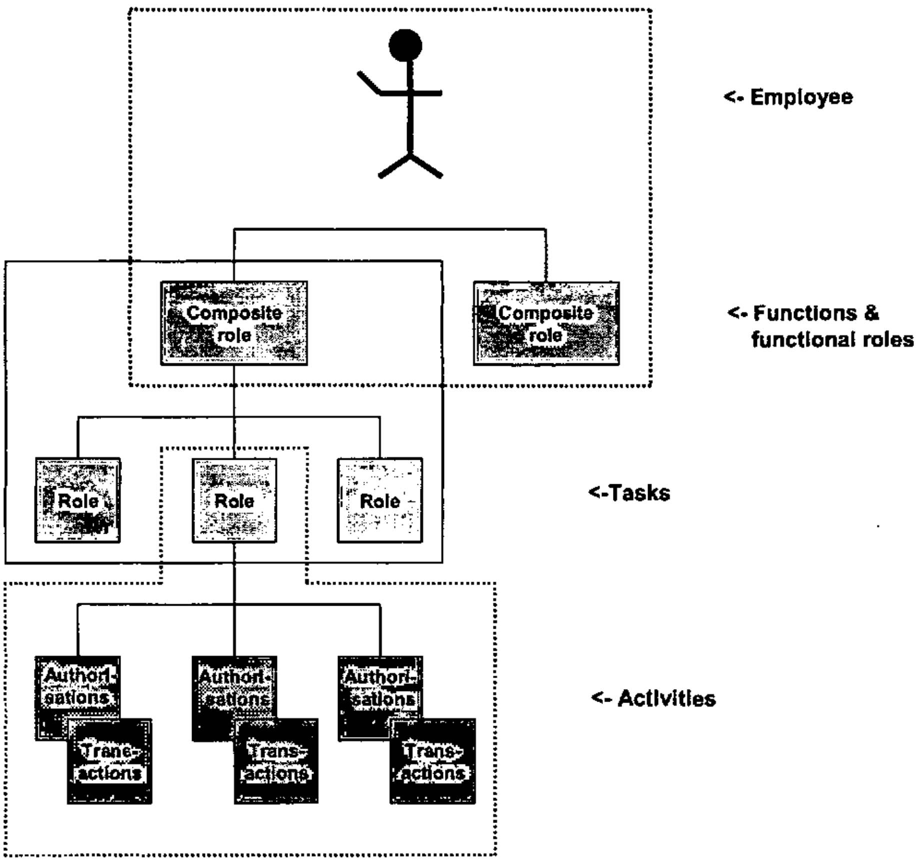


Figure 5.2.2: Relationship between SAP R/3 and the KLPD

Conclusions

- *The implemented authorisation concept is considered to be a very good system.* One of its positive side effects is that the organisation increasingly thinks in terms of functions instead of persons.
- It is important to maintain strict separation of duties between Functional Management (registration function) and Application Management (execution function) with regard to authorisations for SAP R/3. The chosen method makes this quite feasible.
- No action by Application Management is necessary for job changes, so there are many fewer changes.
- The authorisation concept gives auditors and managers a much better view of the current situation than before. In a manner of speaking, the authorisation concept has changed from a 'black box' to a 'white box'.
- The configuration of the Administrative Organisation needs improvement. Beside further improvement of the processes, it is desirable to have a matrix of undesired combinations of functions and tasks.

- New functions should have to be approved by a control body.
- Using authorisations that are valid for only one or a few departments ('area of responsibility') creates much work and restricts the operational flexibility of employees.
- The administration functions related to authorisation must be honed.
- With the change to the SAP R/3 Enterprise release and the new authorisation concept, the KLPD has again reached a 'state of the art' level with regard to authorisations in SAP R/3. The experience gained with the new authorisation concept can also be used in other KLPD information systems and elsewhere within the police organisation.

Recommendations

- Provide good reporting capabilities with regard to granted authorisations.
- Further elaborate the AO procedures, including with regard to Functional Management and Application Management.
- Configure the control function, in addition to the disposition, registration and execution function.
- Investigate the need for restricting authorisations to the area of responsibility of personnel staff members.
- Try to achieve a good balance between technical and procedural measures, instead of attempting to enforce all authorisations by technical means.
- Give further attention to the administration functions.

5.2.3 Management aspects

Introduction

It was decided that this project was to be realised into two phases. Phase 1 consisted of a number of crucial activities and the other activities were carried out in phase 2. I have used a similar division into two phases before. The advantage of this approach is that you are able to concentrate on the truly essential items in phase 1 and are able to deliver the results of phase 1 sooner. The danger can be that phase 2 is executed partly or not at all. However, it is up to the customers (user organisations) to make an effort for the activities in phase 2. Renewed calibration and determination of priorities at the start of phase 2 are important. In this, the Business Case can play an important role. The Audit Department of the Ministry of the Interior and Kingdom Relations were consulted in advance about the arrangement of the phases and corresponding activities.

Business Case

The objective of the POTVIS project is: "*put in order the policy concerning the management and* *security of the SAP R/3 application and the SAP processing organisation by means of implementation of the not (entirely) realised recommendations. In doing so, the SAP R/3 application and the SAP R/3 environment is further professionalized resulting in adequate support of KLPD's Operational Management".*

It is important external parties such as the Audit Department of the Ministry of the Interior and Kingdom relations and the Netherlands Court of Audit have made demands upon the KLPD. The *Programma Verbetering Bedrijfsvoering (Programme Improvement Operational Management) (PVB)* of the KLPD has been assigned to realise matters and to make sure that the KLPD meets the external requirements. With regard to the POTVIS project, this can be viewed as the Business Case. The Business Case was of vital importance during the project because this meant having extra reasons available for obtaining personnel and material resources.

Evaluation

After delivery of the results of phase 1 of the POTVIS project, I wrote an evaluation report. I also wrote the article [jp3] about RBAC. What is special about this article is the fact that different parties (Functional Management, Application Management and Manager Applications) gave their opinion at the end of the article on the authorisation structure as developed and implemented within SAP R/3. On my request, a potter produced POTVIS mugs for all the project officers and staff from the user organisation that were involved in the project. These evaluation activities have definitely contributed to the sense of satisfaction regarding the project.

Methods

- BBNP (Basisbeveiligingsniveau Nederlandse Politie);
- Police Information Security Regulations;
- Role-Based Access Control (NIST); Role-Based matrices;
- Structured Design;
- Structured Testing;
- Structured Documenting.

Staff

The project manager (AvD) was dedicated to the project. Among other things he designed the RBAC model. About fifteen staff members were part-time available for doing different jobs.

Size software

No application software development.

Author	No	Description	Apply to
		Big Hitters	
BH	01	Poor project management	No
BH	02	Deadlines are unrealistic	No
BH	03	Poor communication	No
BH	04	Incomplete/weak definition requirements	No
BH	05	Insufficient involvement of future users	No
		JS+Others	
JS	PUBRC01	Lack of senior management involvement and commitment	No
PN	06	The use of a Business Case results in a higher degree of satisfaction with the project, whilst the satisfaction with the project is very low when no Business Case is used	Yes
PN	10	Technical knowledge is certainly an important skill for managers to have	Yes
PN	14	Evaluating projects is strongly related to the satisfaction with the successful execution of projects	Yes
		Tarek Abdel-Hamid*)	
TAH	01	Software engineering encompasses both the technical aspects of software development as well as the managerial ones {33,47,246}	Yes
TAH	04	There are hundreds of variables that affect software development. Furthermore, these variables are not independent; many of them are related to one another {119,189}. THA constructed *a holistic model* of the software development process	Yes
TAH	10	In a multi-project environment, competition for company resources becomes a significant dimension	Yes
TAH	16	A major challenge to managers is to motivate employees. A mechanism for motivation, which is attracting interest in the software engineering field, is "goal setting" {49}	Yes
TAH	18	Newly hired project members pass through an orientation during which they are less than fully productive {75,243}	Yes
TAH	19	The training of newcomers, both technical and social, is usually carried out by the "old-timers". This is costly because "while the old-timer is helping the new employee learn the job, his own productivity on his other work is reduced {55,74,77}	Yes
TAH	33	Progress in a software project is measured by the number of resources consumed, tasks completed, or both	Yes
TAH	45	A full-time employee allocates, on the average, 60% of his or her time to productive work on the project	Yes
TAH	46	Most software projects start with a smaller core of designers, and as the project develops, the work force slowly builds up to higher levels	Yes
TAH	50	For many years, estimation of software project development time and cost had been an intuitive process based on experience and analogy {200}	Yes
TAH	52	By imposing different estimates on a software project we create different projects. Research clearly indicates that the decisions that people make in projects and the actions they take are significantly influenced by the pressures and perceptions produced by the project's schedule {57,109,125,224,236}	Yes

TAH	54	When the project is perceived to be ahead of schedule, "Parkinson's Law indicates that people will use the extra time for ... personal activities, catching up on the mail, etc. " {49}. This, of course, means that they become less productive	Yes
TAH	57	The "Safety Factor Policy" does achieve its intended objective: more accurate estimates. However, the organisation pays dearly for this	Yes
TAH	60	The policy of allocating project members half-time to the project results in a cost that is about 22% higher	Yes
TAH	66	A project's estimate creates pressures and perceptions that directly influence the decisions people make and the actions they take throughout the project's life cycle	Yes
TAH	77	An integrated approach helps us achieve an overall understanding	Yes
TAH	80	A different schedule creates a different project	Yes
		*) The list is not exhaustive.	

Table 5.2.1: Success and failure factors POTVIS project

Potvis	Score
Complies with functionality agreed	Yes
On time	Yes
Within the agreed budget	Yes

Table 5.2.2: Results POTVIS project

Remarks

The TAH SUFFIs provide more understanding. The SUFFIs TAH/52, TAH/54 and TAH/57 do deserve some explanation. In my opinion, few IT specialists are aware of: "different estimates create different projects" (TAH/52). My choice for a division of the project into two phases has clearly resulted in advantages. It was essential to this project to have finished phase 1 on the 1st of January 2005. This had to do with the Business Case in relation with the Netherlands Court of Audit. As soon as I had drawn up the project planning, one of the (external) project officers remarked that the planning did provide a lot of room for manoeuvre. I answered this that certainly it could be done quicker but that I did know the organisation quite well by then and it would not be excessive to plan extra time because in my opinion it would take rather a lot of effort to get the necessary human resources from the organisation. As proved to be the case. It was striking that TAH/54 and TAH/10 specifically applied to him. I had to talk to him a few times about his responsibility with respect to the POTVIS project.
Project phase 1 was, with some difficulty, completed on January the 1st 2005 and the client (Top Management) was very satisfied. The planning in question had been achieved and the management was able to comply with the Business Case commitment.

In my opinion, TAH/57 is an interesting point of discussion with regard to this project: *The "Safety Factor Policy" does achieve its intended objective: more accurate estimates. However, the organisation pays dearly for this.*

In this project there was:

- a division into phases (TAH/52);
- competition for company resources (TAH/10);
- Parkinson's Law (TAH/54);
- the "Safety Factor Policy" (TAH/57).

The planning could have been more defined: for instance delivering per 1 November 2004. However, it certainly remains to be seen whether this date would have been made. Even the 1st of January 2005 was in danger I think, because:

- the project had come under more pressure;
- I had to put more pressure on managers in order to obtain resources on time.

This could probably have caused an awkward situation that would have been counterproductive.
The built-in "Safety Factor Policy" gave me the opportunity to be able to operate tactfully within the corporate culture in question and thus reaching my goal. More pressure would probably have resulted in extra irritation and would have endangered the final date.

In some environments however, pressure of time is required for obtaining the necessary resources. In that case, a limited "Safety Factor Policy" is applicable.

5.3 CASE: KOLIBRIE (KPN TELECOM)

5.3.1 Introduction

In the period 1999-2001, I worked as an external (part) project leader for KPN Telecom on the project Invoering BaanERP (Introduction BaanERP) (also known as Mecano). During that period, I published the articles jp4 and jp5.

5.3.2 Outline Kolibrie articles

5.3.2.1 Data exchange (interfacing) of applications using a message broker (2001)

Introduction

Business processes are increasingly more supported using ICT supporting information systems. One will often ask oneself these questions: "How many and which information systems does the company / the organisation need?" and "How are these information systems related and what data needs to be exchanged between the information systems?"

The past few decades saw a lot of discussion on these questions. Many intermediate forms came into being between the two extremes. On the one hand, the extremes consisted of a large number of loose information systems that hardly or not at all communicated with each other and on the other hand, attempts were made to arrive at one single comprehensive information system. The first form can be characterised as "island automation". This was very common in the sixties and seventies but even today, a similar type of automation can still occasionally be found. The second form was popular in the seventies. Many computer scientists, in particular those with little actual practical experience, thought that it would be possible to solve all sorts of information problems with a so-called "Management Information System (MIS)". However, the substantial monolithic information systems as envisaged never got off the ground. The right solution consists of a limited number of information systems [jp4]. Important with regard to this is the exchange of data between the information systems. Often these are so-called 1 on 1 links/interfaces. The data of one of the information systems is offered to the other information system and vice versa. Organisations with hundreds of information systems often have a large number of interfaces as well. That is also the case at the Business Unit BedrijfsCommunicatie (Business Communication) (BU BC) of KPN Telecom. The maintenance of all interfaces demands substantial personnel and financial effort. In order to make the management of the interfaces more controllable and in particular for reducing the costs, the BU BC procured the message broker Engin two years ago (in 1999). Using the message broker, a few things have been achieved. Article jp4 discusses the manner in which the message broker is applied and the experiences with it. BaanERP is also connected to Engin and there is exchange of data between some of the other business applications and BaanERP. Article jp5 describes in detail how all this is organised technically.

Problem definition

There are many 1 on I interfaces and high costs are involved in their maintenance. This problem definition is translated into a solution as follows:

- realise as many new interfaces as possible via the message broker;
- during periodic maintenance, replace the current interfaces with interfaces via the message broker, unless there are reasons for not doing so to be approved by the management;
- report on the advantages and disadvantages of using the message broker.

Case

In article jp4, the use of the message broker is explained by means of two information systems, namely M&M007 and FIT.

M&M007

The M&M007 (Meld- en Meetpost 007/Reporting and Testing position 007) information system is used for the registration and settlement of trouble reports concerning telecommunication equipment and systems. With the aid of M&M007, trouble reports are registered, analysed and dealt with. The information system offers support, translating the fault into a service order and uses the appropriate service contract for this, if available.

M&M007 has interfaces with eight information systems, including FIT.

FIT

Using the FIT information system (Field Information Terminal), engineers get their information electronically. To this purpose, the engineers have a mobile workstation and a GSM phone at their disposal (in 2001). FIT consists of two sub systems: FIT Central and FIT Mobile. FIT Central is used for allocation, planning, monitoring of client orders and for creating management information. FIT Mobile is the part of the FIT information system that is installed on the PC's of the engineers. The engineers use FIT Mobile for dealing with client orders.

FIT has interfaces with six information systems, including M&M007.

Interfaces M&M007 and FIT

Between M&M007 and FIT, the following interfaces exist:

- M&M007 offers service orders to FIT which receives these;
- FIT informs M&M007 on the progress of the service orders and provides feedback and ready signals.

Conclusion

BU BC has arrived at the conclusion that using the message broker, provided with an Esperanto facility [jp4], substantially reduces the number of 1 on 1 connections, significantly reduces the costs of realisation and maintenance and enables adequate management of the interfaces.

The Business Unit BedrijfsCommunicatie forcefully pursues its chosen course with regard to data exchange/interfacing of applications.

The conclusion is elucidated by means of a few advantages, disadvantages and guidelines.

Advantages

- a message broker is an application that is fully set up for interface management which makes it easier and cheaper to gear interfaces to each other;
- through application of a message broker, each application is able to communicate with the outside world in its own format. This makes it easier to adjust or replace applications;
- if every application only needs to have one single interface with the outside world, this means that the architecture of the entire "enterprise" will change. The maximum number of interfaces decreases from $n*(n-1)$ to $2*n$;
- knowledge is combined in one single location;
- independent (central) management of the interfaces becomes possible (and is desired);
- it is possible to introduce a high degree of standardisation with regard to transformation;
- connections at data layer level, which endanger the data integrity, are avoided.

Disadvantages

- when the message broker fails, all communication between the applications come to a standstill (single point of failure);
- increase in the interdependency of applications;
- there is a tendency for moving functionality, which belongs in applications, to the message broker.

Guidelines

- in order to streamline the interface management properly it is necessary to set up a management process that is independent from the application. The management process should play a leading role in definition and support of connections and should have sufficient competencies for playing a leading part in interface management. Besides, there should be commitment of all parties involved for having connections running via the message broker;

- the application that is "owner" of the data, decides the record layout of the message;
- the message broker can divide a query into several queries if the answer should come from various different systems. Next, the part answers can be combined into one single answer. When using this functionality, one should makes sure not to include a business functionality in the message broker;
- when two applications exchange data, it is important to know what the purpose of the communication is and which role the applications involved are playing. This determines which actions are carried out and what the contents of the answer message could be. Without this information, a message broker cannot fulfil his role as "intermediate station" correctly;
- the applications should preferably be constructed according to the three-layer-architecture;
- the applications should reckon with the fact that some data from other applications is temporarily not available;
- through building adapters it is possible to simulate a function layer for the benefit of communication;
- connections have to be realised at the level of the function layer. Data layer connections are not allowed;
- applications that are strongly interdependent, must belong to the same domain;
- from a continuity viewpoint it is necessary to take adequate measures for limiting the consequences of a single point of failure;
- for some existing tailor-made connections that include a lot of business functionality, it can make sense not to connect these via a message broker.

5.3.2.2 EAI and ERP with Baan Open World (2001)

Completion EAI at the Invoering (Introduction) BaanERP project

Within the BU BC and the Invoering BaanERP project, EAI (in 2001) is amongst other things completed with *the message broker Engin.* For connecting BaanERP to the other business information systems *Baan Open World* and *adapters* are used.

Figure 5.3.1 shows a (limited) representation of the EAI architecture of the BU BC. The *Engin message broker* acts as a "roundabout for messages". BaanERP communicates via a *Baan-Engin-Adapter* (BEAdap for short) and Engin with other applications such as CKR (Centraal Klanten Register / Central Client Register).

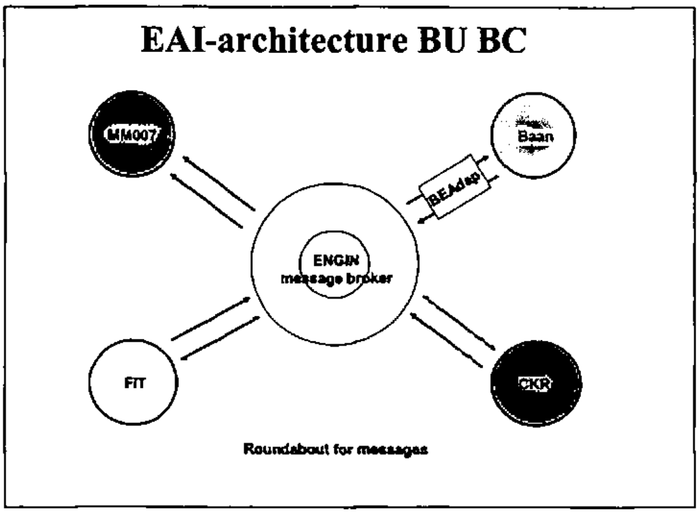


Figure 5.3.1: Message broker Engin and the Baan-Engin-Adapter within the EAI of the BU BC

Baan Open World (BOW for short) is an integration framework allowing Baan Enterprise Solutions applications, including BaanERP, to intercommunicate or to have this communicate with other applications. Within the BU BC's EAI, BOW is used for allowing BaanERP to communicate with the other business information systems via Engin and Baan-Engin-Adapters. BOW is supplementary to middleware and is designed for communicating with a large number of transport protocols and products. Article jp5 includes a detailed description.

The Baan-Engin-Adapter for CKR

CKR is a corporate information system, which is used for registering customer data within KPN Telecom. Other information systems are able to use this centrally managed customer data. To that purpose, CKR has a number of transactions available, such as:

- requesting customer data from CKR;
- applying a reference to a customer. In doing so, a subscription is taken out on transactions. Every day, CKR offers data transactions of the customers on which a subscription was taken out to the subscribers via batch processing;
- removing of a reference.

If the required customer data is not included in BaanERP yet, then this data is requested from CKR and stored in BaanERP by means of an interface between BaanERP and CKR.

How is the interface realised? After carrying out an impact analysis, a Functional Interface Document (FID) is drawn up. Apart from a short introduction of the process in question, this includes data transfer and selection criteria. The FID is the Functional Design of the interface. The FID deviates from the

Functional Design of an "ordinary" information system because an interface includes more than just one single information system. In this case, it is an interface (data exchange) between CKR and BaanERP.

5.3.3 Management aspects

The Invoering BaanERP project was a sizeable project that involved around one hundred project officers. The project included a few sub projects. My role was project leader of the sub project Interfaces. This sub project was given the working title Kolibrie (*K*PN *O*n *L*ine *I*nterfacing with *B*aanERP, *R*esult *I*s *E*ffective) (Kolibrie=Hummingbird). The sub project was executed with about five project officers.

Business Case

For this project, a Business Case was drawn up in advance. The financial section showed that the substantial project costs could be entirely funded from the benefits of the project. Those benefits came into being because it was possible to realise the "completeness of revenues" after completion of the project. In the situation prior to the introduction of BaanERP, activities were sometimes wrongfully left uncharged for. Sometimes, invoices were not sent by mistake and there was insufficient checking on payment. This all changed for the better after the introduction of BaanERP. Furthermore, it became possible to send invoices sooner.

Big Bang

At the start of the Implementation BaanERP project, an external ERP expert, Mr Xbc, of a major audit firm was hired for advising the project leader. During consultations with the sub project leaders he expressly stated that only a "Big bang" introduction of BaanERP was under discussion. Any other solutions were senseless. However, I was of the opinion that a "Big bang" approach was not achievable and would result in a failed project. This point of view was not appreciated. After having read the report, the general project leader who was not present at the meeting, spoke to some sub project leaders and declared the "Big bang" solution as being not practicable.

Communication

The project leader spent a lot of time promoting the communication within the project.
One of the means to this purpose was a weekly "soapbox session". During such a session (a majority of) the project team assembled. The management provided everybody with an insight into the state of affairs and project officers were able to ask questions. Internal employees in particular had trouble asking questions because they were not accustomed to this. During a session like that, the

abovementioned advisor, Mr Xbc, read a paper and gave people the opportunity to ask questions. When an internal employee asked a critical question, the person in question was entirely pulled to pieces. This achieved exactly the opposite of what was intended. The attitude of this advisor, his opinion on the approach ("Big bang" scenario) and his exorbitant hourly rate were all reasons for the project leader to remove him from the project.

The "soapbox sessions" were also utilised for the occasional presentation held by a sub project leader about the state of affairs. I did a few presentations as well. After an extensive presentation and demonstration to do with my sub project, one of the other sub project leaders came to see me and said: "very good, but how can you be that enthusiastic about such an uninteresting subject?"

Made-to-measure

One of the policy starting points was that as little made-to-measure things as possible were allowed to be added to BaanERP. To start with, this rule was strictly obeyed. However, after some time it proved that, under pressure of business request, increasingly more made-to-measure solutions had to be made. That did not just involve a lot of effort for the project team but, in connection with the management of BaanERP, also drew on the future.

Methods

- SDM (System Development Methodology);
- Functional Interface Document, Enterprise Application Infrastructure method, Message broker, Baan Open World (XML, Java, C), adapters;
- Proof Of Concept approach;
- Structured Testing;
- Structured Documenting.

Staff

The project manager was dedicated to the sub project Interfacing. Among other things he designed the EAI architecture together with a BAAN-ERP interfacing specialist. Four other people participated in the sub project Interfaces.

Size software

20,000 lines of code (estimated).

Author	No	Description	Apply to
		Big Hitters	
BH	01	Poor project management	No
BH	02	Deadlines are unrealistic	No
BH	03	Poor communication	No
BH	04	Incomplete/weak definition requirements	No
BH	05	Insufficient involvement of future users	No
		JS+Others	
JS	PUBRC01	Lack of senior management involvement and commitment	No
KY	15	Involve high degree of customisation in application	Yes
PN	06	The use of a Business Case results in a higher degree of satisfaction with the project, whilst the satisfaction with the project is very low when no Business Case is used	Yes
PN	10	Technical knowledge is certainly an important skill for project managers to have	Yes
PN	14	Evaluating projects is strongly related to the satisfaction with the successful execution of projects	Yes
		Tarek Abdel-Hamid*)	
TAH	01	Software engineering encompasses both the technical aspects of software development as well as the managerial ones {33,47,246}	Yes
TAH	05	The behaviour of an individual subsystem in isolation may be different from its behaviour when it interacts with other subsystems {68}	Yes
TAH	09	System complexity grows as the square of the number of system elements; therefore, experience with a small system cannot account for all the things that will have to be done in a large system. Neither will the Experience Method apply to systems of totally different content {23}	Yes
TAH	16	A major challenge to managers is to motivate employees. A mechanism for motivation, which is attracting interest in the software engineering field, is "goal setting" {49}	Yes
TAH	17	All programmers are optimists. They always assume that "This time it will surely run" or "I just found the last bug" {57}	Yes
TAH	19	The training of newcomers, both technical and social, is usually carried out by the "old-timers". This is costly because "while the old-timer is helping the new employee learn the job, his own productivity on his other work is reduced {55,74,77}	Yes
TAH	23	Work force experience level and increases in project familiarity due to learning-curve effects affect the productivity of a software development project {76,234,261}	Yes
TAH	24	The project goals and schedules can play a significant motivational role throughout the life of a software development project	Yes
TAH	29	Newly hired employees are not only less productive on average but also more error-prone than their experienced counterparts {92,189}	Yes
TAH	33	Progress in a software project is measured by the number of resources consumed, tasks completed, or both	Yes
TAH	36	There is a powerful tendency to focus on the highly visible mainline components of the software, and to underestimate or completely miss the unobtrusive components (e.g., help message processing, error processing, and moving data around {49}	Yes
TAH	42	Using the system dynamics model as an experimentation vehicle can provide many benefits to the software engineering community	Yes

TAH	46	Most software projects start with a smaller core of designers, and as the project develops, the work force slowly builds up to higher levels	Yes
TAH	50	For many years, estimation of software project development time and cost had beeo an intuitive process hased on experieoce and analogy {200}	Yes
TAH	52	By imposing different estimates on a software project we create different projects. Research clearly indicates that the decisions that people make in projects and the actions they take are significantly influenced by the pressures and perceptions produced by the project's schedule {57,109,125,224,236}	Yes
TAH	70	Estimates of the fraction of work completed increase as originally planned until a level of about 80-90% is reached. Tbe programmer's individual estimates then increase only very slowly until the task is actually completed" {28}	Yes
TAH	80	A different schedule creates a different project	Yes
TAH	81	An important implication is that a software estimation model cannot be judged solely based en bow accurately it estimates historical projects	Yes
		*) The list is not exhaustive	

Table 5.3.1: Success/failure factors Kolibrie project

Kolibrie	Score
Complies with functionality agreed	Yes*)
On time	Yes*)
Within the agreed Mecano budget	Yes**)
*) Kolibrie project	
**) no specific budget Kolibrie project	

Table 5.3.2: Results Kolibrie project

5.4 CASE: CHARGING METHOD (GAK)

5.4.1 Introduction

Charging out costs is a subject that gets regular attention within Information Technology (IT). However, an adequate solution is not always immediately available. Moreover, a single comprehensive solution does not exist. The article discusses the results of a study into the improvement of charging out the costs involved in making available, management and using of corporate data that is part of the basic register systems. The data is owned by GAK Nederland (in 1998). ASZ Automatisering Sociale Zekerheid B.V. (ASZ Computerisation Social Security BV), which (in 1998) was the regular supplier of computerisation services to the GAK Group, manages the data that is part of the corporate databases and corresponding basic register systems. Within the GAK, there was a crying need for arriving at a different method of charging. People felt the particular need for no longer basing charging out on the traditional performance units but instead basing it on user comprehensible units. But how? This article describes how the Product team Basisregistraties (Basic Registrations) (PB) of ASZ gave this need interpretation.

5.4.2 Outline article Charging method (1998)

Problem definition

The basic register systems are the main area of attention of the PB. Within the GAK's information architecture, the following three basic register systems are an important part of the data infrastructure (corporate data):

- the basic register system *persons* includes the personal data of all policyholders that the GAK maintains relations with;
- the basic register system *employers* contains the data of employers which the GAK maintains relations with or did previously maintain relations with, for administrative purposes of its clients;
- the basic register system *work agreements* contains the current and historical work agreements between persons and employers. The work agreement data remains retrievable until five years after the work agreement has ended.

The data included in the three basic register systems (basic data for short) is stored in three databases. PB acts as an intermediary between the exploitation side and the users and holders side. The costs involved in the exploitation and the use of the basic data (the so-called workload) are charged out by the Beheer Infrastructuur (Management Infrastructure) (BI) department to PB, which next charges these together with the other costs (in particular maintenance costs) on to the holders and users. For the users of the basic register systems, PB is the immediate contact within ASZ. In the past, charging out took place on the basis of performance units which are derived from the workload. In 1997, charging out was based on so-called basic register system licences (BR licences).

Performance units

The concept of workload can be simply described as 'what a system has to process as a result of processing applications'. Workload can be expressed in performance units: CPU time, number of input and output instructions and memory use. In ASZ terms: vups (vax unit of performance), mips (million instructions per second) and gigabytes.

Bottleneck performance units

With regard to the description of the workload, customers of IT services and the technicians in computer centres are often worlds apart. Whilst IT technicians talk about vups, mips and gigabytes, the customers want to talk about their business processes.

BR licences

The data of the basic register systems (BR data) can be accessed using the various different information systems. To this purpose, it is required that authorisation has been given or can be allocated when the data is accessed. Such an authorisation is a BR licence. Working with BR licences has also resulted in a number of bottlenecks [jp6].

Additional requirements

Apart from bottlenecks, there are also additional requirements:

- GAK Nederland BV thinks that ASZ has to work visibly on cost control/cost reduction;
- GAK Nederland BV is of the opinion that ASZ should deliver the charging information such that it can use this information for allocating costs to its products.

Objective

The objective of the study was to arrive at a different method of charging costs, the management (exploitation and maintenance), use of the data collections and applications that are part of the three basic register systems. This should lead to:

- better connection with the customers' business processes;
- more recognisability;
- increased possibility for influencing the volume of the costs as charged;
- control of and (in the case of equal consumption) reduction of the charged costs.

Possible solutions

Charging can take place at different levels. The BSW communication model (Business Service Work) distinguishes three levels of workload: the business processes, the services and the work processes [jp6]. In this study, it was decided to go for so-called functional transactions (services). The functional transactions collection, acts as if it were a communication layer/service layer between the user layer and the data layer. Its starting point is the fact that a large number of information systems communicate with the basic data. From the management point of view, it is undesirable that individual information systems can act immediately on the databases. That would entail individual programmers manipulating and/or changing the databases in their own way which might lead to all sorts of consequences for other users. That is why the manager of the databases needs to have a number of services at his disposal that he can offer to anyone wishing to access and change the basic data (and who is authorised to do so). These services are in the intermediate layer, the service layer/communication layer. The data (base) services may consist of retrieving, updating, adding or removing of one or more record occurrence. The communication services for example, may consist of access security and/or registration of the use of the databases, etc. A great advantage of this approach is that in the case of replacement of the databases by a new collection of databases that are for example supported by another DBMS, it is not necessary to change the information systems. By means of adjustment of the service modules, the users should in fact not even notice this. When the services apart from a technical realisation also have a functional meaning, then charging out may also take place in functional terms. Charging out on the basis of functional transactions (services) provides the users with an overview of the use of the databases. Furthermore, it is important that in doing so, a future-proof method be selected. An important advantage is that ASZ using the already available Transaction Counting System is capable of registering the use of the functional transactions.

Cost sharing model

The cost sharing model includes the rules that ultimately supply the cost prices for fixing the rates. As seen from PB, the main costs are:

- the costs as charged by BI (ASZ's computer centre) for usage and management of the basic register systems;
- the added value as supplied by PB;
- the other costs as charged by other ASZ departments.

The total costs, with added to this the ASZ risk factors, profit factors and policy aspects, are the basis for the fixing of rates (the sales costs). Part of the sales costs will have to be realised through rating of the management and save functions (the sales costs Management). The remainder has to be realised by means of charging out the use of the databases (the sales costs Use). It is necessary to come to an agreement with regard to this division.

Set-up tariff system data use

Requirements tariff system

Although a tariff system is unique to an organisation, the following general quality requirements are applicable: effectiveness, simplicity, recognisability, robustness, fairness, suggestibility and future-proof. A tariff system that is based on charging out the use of functional transactions meets these quality requirements.

Fnnctional transactions

How is it possible to charge out the use of functional transactions? To that purpose, it is necessary to know the weight of the transactions. This can be realised by recording precisely which performance units are part of a certain transaction. However, this will produce problems. It is better to assess the relative weight of the transactions. By allocating a weighting factor (of for example between 1 and 100) to every transaction, it is possible to determine the price of a transaction unit and next of the transactions [jp6].

*The price of the transaction unit = the estimated sales costs Use functional transactions / (sum of all transactions: annual use of a transaction*weighting factor).*

In order to be able to establish the price of the transaction unit, the following data is required:

- the sales costs Data use functional transactions;
- the usage frequencies of the functional transactions;
- the weighting factors of the functional transactions.

Usage frequencies

The usage of functional transactions is registered with the aid of the Transaction Counting System (TCS).

Weighting factors

Functional transactions differ in weight. A simple query transaction is considerably less severe than processing of an update transaction. There are several methods for weighing transactions.

In this case, the weighting factor is based on the number of consulted and updated record occurrences from the logical database. To that purpose, it needs to be established for every transaction what the (average) number of consulted and changed record occurrences is. It seems a sizeable task to show the average number of query occurrences and changed record occurrences of over one hundred functional transactions. However, by means of using the right people this can be realised in just a few days. Next, both are combined into one single weighting factor. In doing so, a change is judged as being four times as weighty as a query. Factor 4 has been decided on after consultation and can be adjusted on the basis of advancing insight.

Other weighting factors also play a part [jp6], the user organisation for example has the opportunity for outlining a policy (to direct) with regard to the use of the functional transactions (services).

Conclusion

The advantages of the selected solution is that because of an adequate realisation and implementation, a substantial contribution is made to the realisation of the ASZ Business plan - Vision and strategy 1996-2000, the bottlenecks as indicated have been removed and the additional requirements have been realised.

Evaluation

The described study has been carried out in a number of thematic steps (see table 5.4.1).

Thematic step	Period	Executor
situation and bottleneck analysis	June/July 1997	external advisor
overall specification solution approach	July 1997	external advisor
development computational model determination tariffs	September 1997	external advisor
elaboration alternative interpretations computational model	October/November 1997	ASZ by agreement with GAK
determination tariffs for 1998	December 1997	ASZ
evaluation first quarter 1998	April 1998	ASZ and GAK
writing article	November 1997, April 1998	external advisor

Table 5.4.1: Thematic steps of the study

How did things go in the first quarter of 1998 using the new method for charging out?

This question was answered in April 1998 by Mr Paul Peltzer, head Product team Basisregistraties (Basic registrations). His answer can be summarised as follows:

- a few users had to get accustomed to the fact that they no longer have insight into used performance units;

- the new method for charging out has caused large shifts between applications with regard to the volume of the charging out costs. Charging out the usage of functional transactions results in a more transparent calculation of the costs;
- the GAK-IT department, the intermediary between GAK and ASZ that determines policy, is very enthusiastic about the new method for charging out because:
  - it has provided GAK-IT with access to better insight into the usage of the IT facilities within the GAK organisation;
  - GAK-IT is able to use the new charging out data as a basis for consultation with ASZ on the way in which the level of costs can be reduced (this is an ambient factor from the ASZ Business plan);
  - discussions on very sizeable differences with regard to invoices are a thing of the past;
- within ASZ, the discussions with supplying departments of PB have increased because PB has also gained more insight into the charging out components;
- ASZ can offer its clients more flexibility because ASZ (with regard to costs) can supply every required (part) product.

The discussion on the relations between service profiles and products have started.
Paul Peltzer: "Over the last few years, we have been charging out according to a method that was mainly built on quicksand. With the 'Van Dijk report', we put a foundation under our policy. The policy has been given a 'face'. The BSW model and the charging out model help us get into the practice of basic cost centres according to service profiles and end user products. Therefore, the study has not just solved our charging out issues but has also resulted in a clear spin-off."

5.4.3 Management aspects

The head of the Product team Basisregistraties (PB) ordered development of the new method of charging. After having worked on this for six months, an internal project group had given back the assignment because the project team was not able to find a solution. The Head of PB was one of the interested parties. During the last period of the project, he reported that the already available results had helped him considerably in tough negotiations with important clients.

Methods

Research project;

- Studying charging methods, ICT organisation ASZ, procedures ASZ, databases ASZ, applications ASZ, BSW communication model;
- Constructing a charging method based on services and weighting factors;

- Structured Testing;
- Structured Documenting.

Staff

The project manager/research engineer was dedicated to the research project. He developed the new charging method. About ten people gave input to the project.

Size software

No application software development.

Author	No	Description	Apply to
		Big Hitters	
BH	01	Poor project management	No
BH	02	Deadlines are unrealistic	No
BH	03	Poor communication	No
BH	04	Incomplete/weak definition requirements	No
BH	05	Insufficient involvement of future users	No
		JS+Others	
JS	PUBRC01	Lack of senior management involvement and commitment	No
PN	06	The use of a Business Case results in a higher degree of satisfaction with the project, whilst the satisfaction with the project is very low when no Business Case is used	Yes
PN	10	Technical knowledge is certainly an important skill for project managers to have	Yes
PN	14	Evaluating projects is strongly related to the satisfaction with the successful execution of projects	Yes
		Tarek Abdel-Hamid	
TAH	05	The behaviour of an individual subsystem in isolation may be different from its behaviour when it interacts with other subsystems {68}	Yes
TAH	08	Two factors significantly influence the initial estimate of the job size: (1) the firm's previous experience; and (2) the general over-all tendency to underestimate the job size {49,223}	Yes
TAH	10	In a multi-project environment, competition for company resources becomes a significant dimension	Yes
TAH	16	A major challenge to managers is to motivate employees. A mechanism for motivation, which is attracting interest in the software engineering field, is "goal setting" {49}	Yes
TAH	36	There is a powerful tendency to focus on the highly visible mainline components of the software, and to underestimate or completely miss the unobtrusive components (e.g., help message processing, error processing, and moving data around {49}	Yes
TAH	50	For many years, estimation of software project development time and cost had been an intuitive process based on experience and analogy {200}	Yes

TAH	63	Variables used in cost estimation tend to be those which are easier to measure, quantify, and estimate, even if they are not the most significant {65}	Yes

Table 5.4.2: Success/failure factors Charging method project

Changing method	Score
Complies with functionality agreed	Yes
On time	Yes
Within the agreed budget	Yes

Table 5.4.3: Results Charging method project

5.5 CASE: TELEPHONY (DUT)

5.5.1 Introduction

In 1993, the Delft University of Technology (DUT) carried out the project "Vervanging Telefooncentrale" (Replacement Telephone Exchange). The objective of the project was to bring a new telephone exchange (PABX) with peripheral equipment into production, which would comply with the functional requirements of the next 7 to 8 years (from 1993) and to offer Telephony Services (TS) as a service for the Computer Centre. A project organisation was formed for the benefit of the 'introduction' phase. Apart from the steering committee and the project coordination, this consisted of the following working groups:

- Marketing, Voorlichting & Opleiding (Marketing, Information & Training) (MVO);
- Inventarisatie & Implementatie (Inventory & Implementation) (I&I);
- Functions (FU);
- Beheer & Bediening (Management & Operation) (B&B).

Replacement of the telephone exchange [jp7] also meant that all the small telephone exchanges on the campus were made defunct and resulted in rollout of new telephones for a large part of the approximately 7,000 connections. Extension of the cabling on the campus and its buildings was also planned.

5.5.2 Outline article IMPALA (1994)

Management

As appeared from literature studies and conversations with advisors and suppliers in the field of private branch exchanges, not a lot had been organised (in 1994) with regard to the management of PABXs (voice communication). Besides, the role of the private branch exchange was changing. In jp7, the control model IMPALA is described. In this model, the theory of the Rotterdam school (IT auditing Erasmus University Rotterdam) played an important part. IMPALA was presented as a framework and could have acted as a basis for the control of the private branch exchange and the Telephony Services that are part of this.

Changing role PABX

The increase in both speech as well as "non speech" communication is also a result of the developments in the field of data communication, facsimile, office automation using personal computers with data communication facilities and the value added services that are increasingly provided as part of the telecommunications network. A number of new developments (1994) are briefly discussed in jp7, namely:

- facsimile;
- paging system;
- automatic call distribution;
- voice mail;
- isdn;
- computer integrated telephony.

The IMPALA control model

Management of telephone exchanges does not just consist of controlling and monitoring but also involves ***control*** in the sense of governing, of technical, organisational, economical and personnel aspects. The model consists of the execution of the following steps:

- step 1: determine what the dependency of the Telephony Services is as seen from the users organisation (GO);
- step 2: assess which general control measurements have been taken and whether these are sufficient for the Telephony Services (TS);
- step 3: assess which TS objects are applicable;
- step 4: assess which quality aspects are applicable;
- step 5: determine the priority of the quality aspects;
- step 6: determine the priority of the TS object/aspect/organisation combinations;
- step 7: develop all relevant TS-object/aspect/organisation combinations according to priority:
    a. fill in the quality aspect;
    b. describe the application focused control measurements;
    c. determine the required management information;
- step 8: set up a Management Control System for each quality aspect.

In this way, it is possible to work according to priority on development of the relevant TS-object/aspect/organisation combinations (step 7) and the Management Control System for the Telephony Services improves all the time (steps 7 and 8 are carried out iteratively).

In the DUT, the following objects are determined for the benefit of the TS:

1 calls;

2 facilities;

3 telephone directory;

4 maintenance;

5 problem management (interruptions, and so on);

6 cabling;

7 change management (assignments/changes);

8 Automatic Call Distribution;

9 voice mail/response;

10 Service Level Agreement;

11 charging out;

12 alarm procedures;

13 operation;

14 paging;

15 Computer Integrated Telephony applications;

16 buying/selling telephones;

17 transfer procedures;

18 emergency plan;

19 control organisation;

20 manual Telephony Services.

In jp7, the following quality aspects are used for starting points:

- effectiveness;
- reliability;
- continuity;
- confidentiality;
- efficiency.

Apart from the abovementioned objects and quality aspects, a further two important things play a part in the control model:

- with respect to the "**regulation**", the KAD model (Kwaliteit van Administratieve Dienstverlening / Quality Administrative Services) plays an important part;
- with respect to the "**control**", the Management Control System plays an important part (monitoring, feedback and response).

5.5.3 Management aspects

Organisation

For the benefit of the 'introduction' phase, a project organisation was formed that consisted of a steering committee and a project group. The steering committee was chaired by the managing director of the Computer Centre (DIR-RC). The head Operations (HOPS) of the Computer Centre and a representative of the suppliers (DIR-LEV) were also part of the steering committee. The project group consisted of the general project manager (APM), an external telecom consultant (ExCons) and four working groups, namely:

- Marketing, Voorlichting & Opleiding (MVO), under supervision of PM-MVO;
- Inventarisatie & Implementatie (I&I), under leadership of PM-I&I;
- Functies (Functions) (FU), under the leadership of PM-FU;
- Beheer & Bediening (B&B), under the leadership of PM-B&B.

My role was that of (external) PM-B&B.

Selection process

The selection process took place under the able leadership of ExCons.
Via a long list procedure a short list procedure was realised, which consisted of 4 suppliers, namely: AT&T, KPN Telecom, Lacis and Siemens. Three large well-known suppliers and small company Lacis. After a careful process, the project group presented a unanimous advice: opt for (the solution) of the Lacis company. Apart from a financially attractive proposition, Lacis had showed thinking along with the client. Lacis was for example the only supplier that without being asked had delivered a document on the applications of Computer Integrated Telephony (CIT) (at Washington University). At that moment in time, CIT was still in its infancy but did show a lot of promise. The other suppliers had not gone to great lengths as far as thinking along with the client was concerned. On the basis of these points (financially attractive and a party that was willing to think along), the Board of Directors was informed that Lacis was selected unanimously.
Only then, did some of the other suppliers wake up and tried to annul this choice through the Board of Directors. Fortunately, DIR-RC was made of stronger stuff. Lacis was chosen unanimously and that was that. The consequence was that some important suppliers broadcasted everywhere that Lacis would certainly not be up to the job and that they were waiting for Lacis' downfall. For Lacis this was all the more reason to be extra motivated and pull out all the stops for getting the job done.

Choice number of sub exchanges

The "Replacement Telephone Exchange" was also quite a task from a technical point of view. One of the points for discussion/decision was whether to go for two or three sub exchanges. ExCons had drawn up a technical proposition that included two sub exchanges. However, PM-I&I wanted three sub exchanges and without any serious proof, argued that this was a much better solution. ExCons indicated that three sub exchanges was going to be more costly and would also be more technically complex. In spite of that, PM-I&I did have his way after consulting with DIR-RC several times.

Drawing up of an inventory

All DUT employees received an inventory form in order to be able to draw up an inventory of the telephony requirements. After all, the new telephone exchange had to be dimensioned on the basis of the data as recorded on the inventory forms. The processing of the inventory forms was one of the tasks of PM-I&I. Within the group, he was asked several times how he was progressing. There was no adequate information provided about the state of affairs. PM-I&I stated that he would make sure that everything was taken care of. The project had started in early 1993 and had to be finished by the end of November 1993. The reason being, amongst other things, that delivery of the telephone exchange involved some ground work to do with cabling and this had to be done before the start of winter. Eventually, the new telephone exchange was dimensioned and ordered on the basis of the information as supplied by PM-I&I.

Crisis team

The APM, the ExCons and the DIR-LEV communicated to the DIR-RC several times that there were serious doubts about the results of the inventory and about a few other matters that came under PM-I&I. One had insufficient insight into the state of affairs. In early October 1993, not even two months before delivery, DIR-LEV told the steering committee that there was no more faith in the procedure and demanded clarity. At that moment, DIR-RC after consultation with the steering committee and APM decided to ask PM-B&B and PM-FU to help the project along as crisis managers.
After an afternoon's consultation, they agreed to do so and started on a few months of working weeks of 7 x 16 hours. The APM was happy about this procedure and soon participated in the crisis team. One of the first things that needed doing was having a good look at the inventory. Soon, this proved to be incomplete, there were still stacks of forms in cupboards that had not been processed. Within a week, the telephone exchange on order had to be expanded. Before mid November 1993, the exchange was expanded twice more. The Lacis company certainly does deserve a huge compliment because they proved to be capable of expanding the new telephone exchange up to three times in a very short span of time. Similar proof of flexibility would probably not have been found in the other three suppliers.

Small telephone exchanges

As indicated in 5.5.1, the plan was to make all (a few dozen) small telephone exchanges on the campus defunct. That almost worked out. Only one single small telephone exchange could not be replaced: the one used by the Board of Directors. In my opinion, this is a bad example of the DUT top management.

Zero hour

On Friday 26 November 1993 at 18.00 hrs, the moment had arrived for disconnecting the old telephone exchange [jp7]. In the auditorium, about 150 staff had gathered. The majority of them got busy under supervision of Lacis rolling out phones and executing cabling activities. On Monday morning, the new telephone exchange was delivered. In the first week, an especially set-up helpdesk was very busy solving problems. After the ISDN problems with KPN Telecom had been solved, the DUT was able to keep going for many years to come. DIR-RC was satisfied after all and said: "In my entire career, this is the most underestimated project I have ever been involved in".

Methods

- Software development: SDM (System Development Methodology);
- Development control model IMPALA: the theory of the Rotterdam school (IT auditing Erasmus University Rotterdam) played an important part;
- Structured Programming (COBOL);
- Structured Testing;
- Structured Documenting.

Staff

The project manager was dedicated to the sub project. Among other things he designed the IMPALA model. Two other staff members were available for doing different jobs.

Size software

10,000 lines of code (estimated).

Author	No	Description	Apply to
		Big Hitters	
BH	01	Poor project management	No
BH	02	Deadlines are unrealistic	No **)
BH	03	Poor communication	No
		Remarks/Observations: The MVO working group ensured excellent communication with the entire university.	
BH	04	Incomplete/weak definition requirements	Yes
BH	05	Insufficient involvement of future users	No
		JS+Others	
JS	PUBRC01	Lack of senior management involvement and commitment	No
		Remarks/Observations: Involvement and commitment were good. However, it should be said that blind faith of a manager in one single employee without	

		remaining critical is very hazardous and, as happened in this case, may lead to a "troubled project".	
JS	RC05	Buyer's funding and/or time-scale expectations unrealistically low	Yes
		Remarks/Observations: DIR-RC: "In my entire career, this is the most underestimated project I have ever been involved in". **)	
JS	RC09	Failure to achieve an open, robust and equitable buyer-vendor relationship	No
		Remarks/Observations: Cooperation between project group and supplier was excellent!	
KY	08	Poor internal communication	Yes
		Remarks/Observations: Within the project team, PM-1&I caused bad communication, which resulted in a "troubled project". **)	
NB	05	Problems escalated too late	Yes
PN	10	Technical knowledge is certainly an important skill for project managers to have	Yes
		Remarks/Observations: In view of the technical complexity of the project, the project manager required technical knowledge. Because there was little knowledge of telephony within the Computer Centre, the ExCons was hired. He graduated from the DUT and was very familiar with the environment.	
		Tarek Abdel-Hamid*)	
TAH	02	Poor management can increase software costs more rapidly than any other factor {261}	Yes
TAH	08	Two factors significantly influence the initial estimate of the job size: (1) the firm's previous experience; and (2) the general over-all tendency to underestimate the job size {49,223}	Yes
TAH	09	System complexity grows as the square of the number of system elements; therefore, experience with a small system cannot account for all the things that will have to be done in a large system. Neither will the Experience Method apply to systems of totally different content {23}	Yes
TAH	10	In a multi-project environment, competition for company resources becomes a significant dimension	Yes
TAH	12	The longer an error goes undetected, the more extensive the necessary rework and the greater the cost	Yes
TAH	13	Inadequate planning is the primary reason for loss of control on many computer programming projects {208}	Yes
TAH	16	A major challenge to managers is to motivate employees. A mechanism for motivation, which is attracting interest in the software engineering field, is "goal setting" {49}	Yes
TAH	28	Design errors in the early design phase are generated at a higher rate than are coding errors {165}	Yes
TAH	31	We assume that undetected errors will become either "Active errors" (errors that produces more errors) or "Passive errors". Because design specs are the blueprints of the system's code, any errors in design will get translated into coding errors	Yes
TAH	35	As a software project develops, project members often realise that they have under-estimated the number of tasks that constitutes the software system being developed {61}	Yes

TAH	36	There is a powerful tendency to focus on the highly visible mainline components of the software, and to underestimate or completely miss the unobtrusive components (e.g., help message processing, error processing, and moving data around {49}	Yes
TAH	37	In some cases the management becomes increasingly willing to "pay any price" necessary to avoid overshooting the "Maximum Tolerable Completion Date". In such cases, management is often willing to hire more people	Yes
TAH	42	Using the system dynamics model as an experimentation vehicle can provide many benefits to the software engineering community	Yes
TAH	44	By using a model of a complex system, more can be learned about internal interactions than would ever be possible through manipulation of the real system	Yes
TAH	50	For many years, estimation of software project development time and cost had been an intuitive process based on experience and analogy {200}	Yes
TAH	64	Our model results indicate "adding manpower to a late software project makes it more costly, but *not necessarily later*"	Yes
TAH	65	A different distribution of estimated effort among a project's phases creates a different project	Yes
TAH	68	When a software development project overruns its schedule there is an apparent cause: the project was poorly estimated	Yes
TAH	81	An important implication is that a software estimation model cannot be judged solely based on how accurately it estimates historical projects	Yes
		*) The list is not exhaustive **) SUFFI BH/02 = No. At the start of the project, the very experienced ExCons had stated that the desired mid November 1993 deadline was realistic and therefore feasible. The management underestimated the amount of work that had to be shifted in the year 1993. That was considerably more than initially expected. This besides the bad communication between PM I&I, which resulted in a 'troubled project' (see also section 9.5). The project was delivered on time!	

Table 5.5.1: Success/failure factors Telephony project

Telephony	Score
Complies with functionality agreed	Yes
On time	Yes
Within the agreed budget	No *)
*) but it did not exceed the build cost by more than 50%	

Table 5.5.2: Results Telephony project

5.6 CASE: OKAPI (UoA)

5.6.1 Introduction

In 1991, over 28,000 students were registered with the University of Amsterdam (UoA). Apart from teaching, this university also carries out research. The UoA processed its information in the eighties and nineties using the SARA computers. The administrative information systems used to run on a CDC computer until 1 July 1992. The information system of the Dienst Bouw en Huisvesting (Building and Housing Service) (DBH) was also run on the CDC. In the year 1990, SARA decided to dispose of the CDC computer as per 1 July 1992. A new large IBM computer was available for supporting these applications. For that reason, the information system of the DBH had to be renewed and to be available on the IBM computer on the 1st July 1992.

5.6.2 Outline OKAPI article (1994)

Introduction

For the education and research at the UoA to run properly, buildings are needed amongst other things. The Dienst Bouw en Huisvesting (DBH) translates and carries out the building and housing policy as decided by the university administration. In 1992, the DBH put the new information system OKAPI (*O*nline *KAPI*taaldienst (capital service)) into operation. Using OKAPI the building projects are financially supported and monitored. OKAPI was realised using the IDMS DB/DC database management and data communication system and the ADS/online development tool.
During construction the objectives were:

- the new information system should be available on the 1st July 1992;
- the users have to be able to navigate quickly and without any problems between the many display images;
- data input and monitoring should meet very high requirements with regard to reliability;
- users can/may only carry out authorised actions;
- data from the old file needs to be converted to the file of the new application.

The application

In the financial building administration the construction of new buildings and renovation projects are registered. The administration is supported by OKAPI. In the OKAPI database a large amount of data is registered: project data, tasks, supplemental commissions, reduction commissions, approximate estimates, commissions from approximate estimates, provisional commissions, terms, invoices, payments, allocated budgets, working budgets, financing sources, bank guarantees, project officers, providers, general ledger accounts and cost categories. On the one hand, OKAPI is an information

system for the financial building administration (including budgetary control); on the other hand, it is a sub ledger administration/accounts payable section.
The online part of OKAPI consists of 132 display images that are divided over five sub systems, namely:

- consult/change projects;
- consult/change sundries, such as: buildings, bank accounts, cost categories, commission groups, providers, project officers, collective entries payments, financing sources and ledger accounts;
- consulting creditors;
- report;
- system management.

The OKAPI users can be divided into five categories (roles). With the aid of an authorisation table it is indicated who, based on the role that they play, has access to which display image and what the competencies are (role based authorisation). Project leaders can also authorise users with the role of "project officer" within their project on the basis of the role that they play within the project (two phase role-based authorisation).

Realisation
In order to be able to meet the set requirements, a number of arrangements have been made, including:

- within OKAPI over 800 (error) messages have been stored for assisting the user entering data correctly;
- the sub system report includes extensive possibilities for the users for providing batch jobs with parameters, starting these and for tracking these;
- an "OKAPI monitor" has been designed and realised. The monitor enables the user to go quickly through the required display images. Furthermore, the OKAPI monitor takes care of the monitoring on the authorisation at display image level.

5.6.3 Management aspects

Steering committee/project group
The steering committee consisted of the Head of the BIA department (Bestuurlijke Informatie en Automatisering/Administrative Information and Automation) and the Head of the finances department of the DBH. The steering committee cooperated well with the project group, which consisted of an external part time project leader (Aart van Dijk), 3 internal designers/programmers and an internal system designer/programmer for the Conversion sub system. The introductory talk was interesting.

During this meeting, the intended internal staff asked: "What are you doing here as a project leader, nothing ever comes of anything here?". My answer was: "As you all know I have gained a lot of experience within universities and in my experience things are totally different. When you go to any university and stand in the corridor shouting: who will help me, then it usually remains fairly quiet. However, if you take the initiative and are prepared to get your hands dirty and do a lot of the work yourself, then usually an awful lot is possible!". We got to work and after a few weeks, we slowly developed a fine team that did achieve an excellent result.

Key user

The key user is expressly involved in the project. I did not just make a functional design but also a prototype. By means of this prototype, the users were able to get a good idea of the future system. The key user was very critical. Not just because he had already worked in the financial building administration for 25 years and therefore had in-depth knowledge of the subject matter but also because the existing information system that was running on the CDC, had been produced by him. It was not a professional system and he was the only person able to work it. Nevertheless, I had greatly appreciated the initiatives that he had taken. Although he was no information scientist, he still managed to make a system that worked for him. With the new OKAPI information system, his "baby" was to disappear and that did produce a bit of a lump in his throat. The presentation of the functional design/prototype to the management of the University Office and the DBH was taken care of by the project leader in close cooperation with the key user. This was greatly appreciated by everybody.

Conversion

During construction, a lot of time was also spent on the sub project Conversion. Apart from conversion software, which was to be used for loading the IDMS database in the last week of June 1992, several programs were made for being able to analyse the old data files and where necessary amend these. This was a tricky job because when analysing the old data files it was necessary to reckon with the improper use of fields by various users in the course of time. The sub project Conversion was executed under my leadership by someone who just got his degree in Information Sciences at the UoA. After having been steeped in theory for years, he was able to indulge in the "unruly" practice.

Functionality

In spite of all those involved being fully dedicated, the available manpower proved insufficient to fully finish the project before 1 July 2002. Nevertheless, at this moment in time all the essential online functions and the conversion software were ready. Over the period July-December 1992, things like the sub system Report, the annual work procedures and rounding off of the documentation were realised.

Project evaluation

In jp8, a brief project evaluation is included as well. This states: "OKAPI is an example of a successful realisation, although the lead time was a little longer than expected".

Methods

- SDM (System Development Methodology);
- HIPO, Screen image dialogue structures, normalisation of data, Bachman-diagram, prototyping with key user;
- Structured Programming (IDMS, ADS/online, COBOL);
- Structured Testing;
- Structured Documenting.

Staff

The part time project manager was dedicated to the project. Among other things he designed the functional design. Four other staff members were available for doing different jobs.

Size software

90,000 lines of code (estimated).

Author	No	Description	Apply to
		Big Hitters	
BH	01	Poor project management	No
BH	02	Deadlines are unrealistic	No
		Remarks/Observations: From the start of the project, it was clear that the date of 1 July 1992 was a final date	
BH	03	Poor communication	No
BH	04	Incomplete/weak definition requirements	No
BH	05	Insufficient involvement of future users	No
		JS+Others	
JS	PUBRC01	Lack of senior management involvement and commitment	No
		Remarks/Observations: Cooperation between steering committee and project group was excellent	
PN	10	Technical knowledge is certainly an important skill for project managers to have	Yes
		Remarks/Observations: In view of the technical complexity of the project and the limited project group, the project manager required technical knowledge. The OKAPI monitor for example was designed by the project manager.	

		Tarek Abdel-Hamid*)	
TAH	01	Software engineering encompasses both the technical aspects of software development as well as the managerial ones {33,47,246}	Yes
TAH	16	A major challenge to managers is to motivate employees. A mechanism for motivation, which is attracting interest in the software engineering field, is "goal setting" {49}	Yes
TAH	17	All programmers are optimists. They always assume that "This time it will surely run" or "I just found the last bug" {57}	Yes
TAH	19	The training of newcomers, both technical and social, is usually carried out by the "old-timers". This is costly because "while the old-timer is helping the new employee learn the job, his own productivity on his other work is reduced {55,74,77}	Yes
TAH	24	The project goals and schedules can play a significant motivational role throughout the life of a software development project	Yes
TAH	33	Progress in a software project is measured by the number of resources consumed, tasks completed, or both	Yes
TAH	36	There is a powerful tendency to focus on the highly visible mainline components of the software, and to underestimate or completely miss the unobtrusive components (e.g., help message processing, error processing, and moving data around {49}	Yes
TAH	45	A full-time employee allocates, on the average, 60% of his or her time to productive work on the project	Yes
TAH	50	For many years, estimation of software project development time and cost had been an intuitive process based on experience and analogy {200}	Yes
TAH	70	Estimates of the fraction of work completed increase as originally planned until a level of about 80-90% is reached. The programmer's individual estimates then increase only very slowly until the task is actually completed" {28}	Yes
TAH	77	An integrated approach helps us achieve an overall understanding	Yes
TAH	80	A different schedule creates a different project	Yes
		*) The list is not exhaustive	

Table 5.6.1: Success/failure factors OKAPI project

OKAPI	Score
Complies with functionality agreed	Yes
On time	Yes/No*)
Within the agreed budget**)	----**)
*) all essential online functions and conversion functions were ready on time. Reports, annual procedures and documentation at a later date	
**) no specific project budget, however a total budget for conversion from CDC to IBM	

Table 5.6.2: Results OKAPI project

5.7 CASE: GIRAF (DUT)

5.7.1 Introduction

In 1970, the (batch)system Bibliosystem was developed at the DUT [see par. 5.10 and jp14]. In 1976, the Computer Centre of the DUT developed the online retrieval system BIBINFO. This system fitted in with Bibliosystem. One of the reasons for developing BIBINFO was the international AICA congress that the Computer Centre organised in 1976. Bibliosystem and BIBINFO were developed for a limited target group. The systems were developed using methods and tools that were modern in that day. The big success of these systems within the DUT and the continuous demand for similar facilities made the management of the Computer Centre in 1982 decide to have a new information retrieval developed, which was aimed at a broader target group. It was to replace the Bibliosystem and BIBINFO system and had to be very user-friendly. This new system was the software system GIRAF (General Information RetrievAl Facilities).

5.7.2 Outline GIRAF articles

5.7.2.1 GIRAF (1984)

Starting points

From the success of the Bibliosystem and BIBINFO systems, the conclusion was drawn that the philosophy on which these systems are based, is a good philosophy as far as the DUT is concerned. The first condition that had to be fulfilled by GIRAF was that this philosophy was maintained. This meant amongst other things that GIRAF had to become a simple and effective information retrieval system (*requirement 1*). The data unit in the old systems was the reference. Access to an *individual* reference took place via the reference number or the author. Access to a *collection* of references was only possible via a code system. This code system consisted of around ten attributes with for every attribute that was a collection of values. Per attribute, a value from the appropriate collection of values could be attributed. This way, a user was able to attribute characteristics and characteristic values to a reference. The selection of a sub collection could be realised by stating the attributes, in which one was interested for the selection in question. Per attribute, the collection of values, as valid for this selection, was stated. By means of AND and OR relations at attribute level it was possible to construct the desired sub collection. Because GIRAF had to be made for a broader target group, it had to be more generally usable as compared to the old systems (*requirement 2*). From this, also as a result of the requests from users, the conclusion was drawn that the code system had to be broader *(requirement 3*). The DUT is a multiform organisation that includes many independently operating groupings. Some of those groupings wish to realise (parts of) their information in an isolated manner (for example PhD students) *(requirement 4*). Other groupings do like to cooperate with grouping within or outside the DUT. For those groupings, it is necessary to be able to build, consult and/or change files from different locations and with the aid of online processing *(requirement 5).* GIRAF needs to be able to be applied for the benefit of many information systems. To that purpose, GIRAF has to be offered in the shape of an off-the-peg system. However, the set-up had to be such that it is possible to adjust it without too much effort. That way, GIRAF can functions for some users as a *tailor-made-off-the-peg system (requirement 6).* Foreigners also work and study at the DUT. For that reason, it is desirable that GIRAF can also work using other languages. The system has to be set up such that other language versions can be implemented quickly at a later stage *(requirement 7).* The users of Bibliosystem and BIBINFO have to be able to transfer to the new system in a flexible way *(requirement 8).* The Computer Centre of the DUT selected IDMS DB/DC as a support system for database management and data communication activities. GIRAF had to be developed using IDMS DB/DC *(requirement 9).*

Sub systems

The GIRAF software system is divided into four sub systems, namely:

- sub system 1: conversion/building database;
- sub system 2: online activities;
- sub system 3: standard batch tools;
- sub system 4: user-dependent applications.

Using sub system 4, certain user categories are able to extend GIRAF with their "own" online system. This own online system has to comply with certain preconditions and can be included in GIRAF in a simple way. Paragraph 5.7.2.2 further discusses this subject matter.

Within the DUT, GIRAF has been utilised for various applications. The University Office (Documentary Information management) has been using GIRAF for years already, has registered some hundreds of thousands of letters, and monitored how these were processed. The election system for the DUT is also based on GIRAF. Outside the DUT, GIRAF was procured by some universities and large institutions in the Netherlands and Belgium (situation in 1994).

5.7.2.2 ADS/ONLINE (1984)

Introduction

Paragraph 5.7.2.1 provides an outline of the requirements that have to be met by the GIRAF software system. One of the requirements was that GIRAF had to be realised using IDMS DB/DC. The appropriate development tool for IDMS is the Application Development System/online (ADS/online

for short). ADS/online has to be viewed as a TP (TeleProcessing) application under IDMS/DC. Centre stage is the concept dialogue. With the aid of ADS/online it is possible to develop dialogues. In a dialogue of ADS/online the display image takes centre stage. A display image is described using a map. A map is a type of table in which all fields of a certain display image are defined with a symbolic name. The fields consist of text fields and input/output fields. The last mentioned fields are linked to one or more records, the so-called map records. Apart from display image (or map), an ADS/online dialogue also has a so-called pre map process and response processes. A pre map process is a program that, after the dialogue in question is activated, is executed before the map (the display image) is made visible on the screen. After the user of the dialogue, which is part of some application or other, has reacted to what he has seen on the screen, a response process as part of the dialogue is started. The selection of the response process is also determined by means of an action that is taken by the user. Pressing the "enter" key for example will possibly activate a different response process than pressing of the "clear" key. Apart from map records, processes often need work records. To sum up, it can be stated that an ADS/online dialogue (usually) consists of: a pre map process, a map with corresponding map records, one or more response processes, sub scheme records and work records.
A dialogue is usually part of a collection of dialogues. In that case, this collection of dialogues is the online part of an application. In jp10, the coherence and the communication between dialogues is discussed. Other subjects that will come up for discussion are:

- dialogue structure diagrams;
- dialogue communication diagrams;
- online mapping;
- conversational programming versus pseudo conversational programming;
- an extensive description of the designed "cassette system";
- multiple databases;
- prototyping in relation with SDM;
- an in-house developed index method.

The in-house developed cassette system is also discussed in jp10. The in-house index method was developed because the index method as delivered by the supplier did not work properly at the time. The index method was presented during an international congress in the United States of America. A number of the companies present did (temporarily) adopt the index method.

5.7.3 Management aspects

Project

The big success of these systems within the DUT and the continuous demand for similar facilities made the management of the Computer Centre in 1982, which was also advised to do so by the builder of these systems, decide to have a new information retrieval system developed, which was aimed at a broader target group. It was to replace the Bibliosystem and BIBINFO system and had to be very user-friendly. This new system was the software system GIRAF (General Information RetrievAl Facilities). The project group had to defend itself regularly during the project because some managers wished to put the priorities with other projects. After GIRAF came into use, the Manager of the Computer Centre said the following: "This system had grown against all odds and was successfully completed".

Innovative

The development of GIRAF was, also from a technical point of view, a challenging task. A cassette system was developed. Using this cassette system, it is possible to replace one or more dialogues specific to one single user without having to adapt GIRAF. Using sub system 4, certain user categories are able to extend GIRAF with their "own" online system. This own online system has to comply with certain preconditions and can be included in GIRAF in a simple way without GIRAF having to be adjusted. Users can work in different languages. At start-up, a user can select in which language he wishes to work. Of course, the display images and messages in the language in question need to be loaded. At an IDMS congress in Copenhagen, the American supplier announced that with the new multi mapping facility the wish of several European countries for being able to work in different languages was fulfilled. Unfortunately, all (error) messages still appeared in English. Personally, I did not consider a German dialogue with English error messages to be very user-friendly. At the congress, I was able to demonstrate by means of a number of sheets that I had brought with me, that the DUT already has a multi language version operational in which the displays and (error) messages were shown in the same language. In German display images, the error messages were also in German.
Because of GIRAF's flexibility, the management was able to avoid the building of a number of tailor-made systems.

Use outside the DUT

Apart from a number of applications within the DUT, GIRAF has also been put into operation at the UoA for amongst other things, the information system for the elections and at the Rijksuniversiteit Leiden (RUL), that used it to create a temporary provision for the thesaurus system of the Library. After the publications and a presentation at an international congress there was also some interest from outside the Netherlands. A large institution such as the Vlaams Economisch Verbond in Antwerp for example, used GIRAF for a number of years for dealing with its mail. After the publications came out, a few companies in the Netherlands and Belgium introduced the cassette system for all IDMS applications.

However, the DUT did not wish to carry out any activities outside the DUT and also refused to take on any maintenance obligations. For that reason, the DUT entered into a contract with the company Avédé-Info BV, which has taken on these activities. The cooperation between these parties was excellent. The Manager of the Computer Centre ir. J.C. Zuidervaart deserves a special compliment because he made many things possible.

Methods

- SDM (System Development Methodology);
- HIPO, Screen image dialogue structures, normalisation of data, Bachman-diagram, decision tables;
- Structured Programming (IDMS, ADS/online);
- Structured Testing;
- Structured Documenting.

Staff

The project manager was dedicated to the project. Among other things he designed the functional design and architecture of GIRAF. About three staff members were available for doing different jobs.

Size software

160,000 lines of code.

Author	No	Description	Apply to
		Big Hitters	
BH	01	Poor project management	No
BH	02	Deadlines are unrealistic	No
BH	03	Poor communication	No
BH	04	Incomplete/weak definition requirements	No
BH	05	Insufficient involvement of future users	No
		JS+Others	
JS	PUBRC01	Lack of senior management involvement and commitment	No
PN	10	Technical knowledge is certainly an important skill for project managers to have	Yes
		Remarks/Observations: In view of the technical complexity of the project and because of the fact that the project manager was the initiator of the project, the project manager did need technical knowledge.	
		Tarek Abdel-Hamid*)	
TAH	01	Software engineering encompasses both the technical aspects of software development as well as the managerial ones {33,47,246}	Yes
TAH	04	There are hundreds of variables that affect software development.	Yes

		Furthermore, these variables are not independent; many of them are related to one another {119,189}. THA constructed *a holistic model* of the software development process	
TAH	05	The behaviour of an individual suhsystem in isolation may be different from its behaviour when it interacts with other subsystems {68}	Yes
TAH	08	Two factors significantly influence the initial estimate of the job size: (1) the firm's previous experience; and (2) the general over-all tendency to underestimate the job size {49,223}	Yes
TAH	10	In a multi-project environment, competition for company resources becomes a significant dimension	Yes
TAH	16	A major challenge to managers is to motivate employees. A mechanism for motivation, which is attracting interest in the software engineering field, is "goal setting" {49}	Yes
TAH	33	Progress in a software project is measured by the number of resources consumed, tasks completed, or both	Yes
TAH	44	By using a model of a complex system, more can be learned about internal interactions than would ever he possible through manipulation of the real system	Yes
TAH	50	For many years, estimation of software project development time and cost bad been an intuitive process based on experience and analogy {200}	Yes
TAH	52	By imposing different estimates on a software project we create different projects. Research clearly indicates that the decisions that people make in projects and the actions they take are significantly influenced by the pressures and perceptions produced by the project's schedule {57,109,125,224,236}	Yes
TAH	76	QA policy does have a significant impact on total project cost	Yes
TAH	77	An integrated approach helps us achieve an overall understanding	Yes
		*) The list is not exhaustive	

Table 5.7.1: Success/failure factors GIRAF project

GIRAF	Score
Complies with functionality agreed	Yes
On time	Yes*)
Within the agreed budget	----**)
*) a lot of hard work was done but there was no hard time line	
**) no specific project budget	

Table 5.7.2: Results GIRAF project

5.8 CASE: AUBID (DUT)

5.8.1 Introduction

The AUBID project concerns the computerisation of the Library of the Delft University of Technology (DUT). On this project, which ran over the period 1977-1980, I published two articles in the trade magazines [jp11, jp12]. The next paragraph will include brief summaries of these articles. In paragraph 5.8.3, management aspects will come up for discussion.

5.8.2 Outline AUBID articles

5.8.2.1 Delft students request library books from computer (1979)

Introduction

The Library of the DUT has existed since 1842 and is a scientific library, which apart from the function as a library for the DUT also fulfils the national function of Central Technical Library. Of course the collection, no matter how extensive this is (in 1979: approx. 554,000 parts, of which 332,000 books and 222,000 magazine portfolios), includes just part of the world literature in technical and related fields. In order to provide the user of the library with a maximum of assistance, close cooperation with other libraries both at home and abroad was established. The Delft library received a lot of international attention during the sixties because of the unique system that customers could use for requesting books using the so-called "bibliofoon". The bibliofoon created a direct connection between visitor and book stockroom. In 1977, the Library heard of an integrated library system that was developed in Dortmund, which was possibly also interesting for Delft . This concerned the Dortmund Bibliotheks-system (DOBIS). As soon as a study group of the Library had investigated the library technical aspects of DOBIS, the Computer Centre of the DUT was asked to find out whether it was possible to implement DOBIS on the computer system of the Computer Centre. After a preliminary study [Van Dijk & Looijen 1977], it was possible to take the affirmative view on this question.

Intermediary request system

The DOBIS system is a product of the Gesamthochschule Dortmund. In the Gesamthochschule Dortmund, the following institutions participate: the Universitaet Dortmund, the Paedagogische Hochschule Ruhr and the Fachhochschule. DOBIS was created for computerising the three university libraries, as well as 25 departmental libraries. Specific to this is the fact that the books owned by the collective libraries are considered as one single entity, whilst the various book collections are stored decentralised. The Dortmund library is an "open library". This means that the person who requests the books, gets these out of the bookcase himself and reports to the lending counter with the books and the library staff take care of the formalities.

In Delft, the situation is clearly different because this is a library of the closed type. The (potential) customer searches for the books that he or she needs in the catalogues, makes a note of the book numbers (1977) and using the bibliofoon, enters the required book numbers and waits until his/or request has been dealt with. This piece of dialogue was not (yet) available in DOBIS and had to be developed by the DUT. With regard to this, it was important that this part of the dialogue has to be very user-friendly because visitors to the Library on average had no experience whatsoever with display terminals (1977). Before the construction of the "Intermediair aanvraagsysteem" (Intermediary Request system) was started, a number of requirements were formulated to which the system had to comply. The following functions had to be available in the system:

- generation of a status file, which contains the booking status of each book. Such a booking status could be: available, lend out, not available, and so on;
- record the number of copies of books that have duplicates;
- process request for books and by means of the requests updating of the status file;
- of the requested books (also including the duplicates) that appear to be available, print the book number and any other possibly relevant information on the matrix printer of the book stockroom in question;
- process books that are handed in, which for example entails having to update the status file;
- entering new book numbers (including duplicates) into the status file;
- setting up a statistics file with regard to the requests that were made and produce lists.

These requirements were supplemented by requirements that emerged from interviews with library staff. The intermediary request system was set up by a small project group composed of staff from both the Library as well as the Computer Centre.
Early 1979, it was delivered and after an extensive test period, the intermediary request system was put into use on the 31st of October 1979 [see photos jp11]. The intermediary request system functioned for over 10 years, before a similar functionality was added to DOBIS.
In the next paragraph, the technical side of the intermediary request system will be further discussed.

5.8.2.2 Experiences with the CICS/VS Command-level (1980)

Introduction

In 1968, IBM announced the Customer Information Control System (CICS). This message control system is designed for dealing with transactions in a database / data communication environment. CICS is an interface between the operating system and a number of message processing programs. In the first version of CICS, the programmers did have to know a computer language but also needed to be well informed of a number of internal CICS affairs because manipulating using pointers, control-blocks, and so on was an integral part of the application software. In 1977/1978, apart form the existing MACRO-level version of CICS, a Command-level version of CICS was announced, which meant that the manipulation using control-blocks, pointers and so on was a thing of the past. At the start of 1978, the DUT Computer Centre procured CICS/VS for use in the Library. In jp12, a number of problems concerning the application (intermediary request system) and the solutions as constructed in Command-level CICS/VS will be discussed.

CICS/VS in relation to the application

The application concerns the intermediary request system that is summarised in paragraph 5.8.2.1. In jp12, a large number of subjects will come up for discussion:

- starting up the application;
- segregation of functions

  The intermediary request system has four (functional) types of terminal:
  function type 1: terminal for hall requests with Dutch language communication
  function type 2: terminal for hall requests with English language communication
  function type 3: terminal for information counter or lending counter
  function type 4: terminal for PTT requests;
- stockroom printers

  Jp12 describes a method that was developed for identifying problems with printers that do not function or do not function properly. This method (the so-called "WEKKER-task" or "ALARM CLOCK-task" method) was developed within the project. Publication in the magazine "Informatie" in April 1980 resulted in the "WEKKER-task"-method being implemented in many businesses;
- the switching of stockroom printers

  Standard, printer 1 is allocated to the book stockroom 1, book stockroom 2 receives the requests and messages via printer 2, and so on. Sometimes it is better to connect another printer to a particular book stockroom, for example when a printer is broken; it is possible to send the requests temporarily through another printer. From a business organisational viewpoint it is also sometimes advisable to connect several stockrooms with one single printer, for example when one member of staff has to work in several stockrooms at the same time during quiet times;
- sending of terminal messages;
- the use of the CICS/VS master-terminal-transaction;
- the chaining of CSMT-transactions;
- recovery/restart;

- the Common Work Area;
- emergency restart;
- Transient-Data-Files;
- dead-lock situations.

5.8.3 Management aspects

HSBI

In late 1977, the Computer Centre was asked to realise the intermediary request system (phase I of AUBID). The HSBI (Hoofd Stuurgroep Bestuurlijke Informatievoorziening (Head Steering committee Managerial Information provision)) appointed me project leader. The HSBI managed all projects regarding information provision. In 1976, the ambitious DAISY project was started with the assistance of Pandata. All import new information systems (PERSIS, SIS, etc) had to be developed within this project. Virtually all available resources (personnel and material) were allocated for the benefit of DAISY. Mid 1976, PERSIS (personnel information system) and SIS (student information system) were started. AUBID did not come under DAISY. On the 31st of October 1979, the intermediary request system was officially put into use [see photos jp11]. During the official conveyance of the intermediary request system by the HSBI to the management of the Library, vice-chairman drs. N.R. van Dijk expressed his surprise but also his appreciation with regard to the fact that, whilst DAISY in spite of all priorities and means still did not have a single information system operational, the AUBID project team phase 1, with ir. M. Looijen from the HSBI managed to build and implement such a complex and sizeable information system successfully. The project team was happy with this sincere compliment.

Computer centre

Within the Computer Centre, the assignment was carried out by the department Informatiesysteem Ontwikkeling (Information System development) (ISO), under leadership of ir. M. Looijen (remark: prof.dr.ir. M. Looijen was appointed as professor at the DUT a few years later and has by now reached emeritus status [Van Dijk 2001]). I was appointed as project leader by the HSBI at the request of the Computer Centre. The management of the Computer Centre was very motivated to realise the chosen solution.

Project group

The project group consisted of 5 members of staff, including two Library staff.

The project group had a challenging and complex task:

- it was the first online information system at the DUT that had to support the work processes of a large group of employees on a daily basis;
- it was the first information system within the DUT that was to use transaction monitor CICS/VS and the DUT was one of the first institutions in the Netherlands that was to use the CICS/VS COMMAND-level;
- the introduction of the intermediary request system meant that a number of staff had to adopt a different working method. In fact, it involved a reorientation of the administrative organisation;
- within the Library, a sizeable SNA-network had to be set up;
- the pressure for arriving at adequate results was high, whilst human resources were very limited. One day, when I remarked to Looijen that certain things (complexity, high expectations and minimum resources), did not balance, he answered: "The most beautiful diamonds are cut under the highest pressure!" [Van Dijk 2001].

The project group was highly motivated for tackling the challenging task and to realise it successfully.

Library

Within the Library, there was a vast difference of opinion between the management and the management of those departments involved:

- the Librarian (manager) did not really want any computerisation activities in his Library. When, after eighteen months of hard work, the SNA network was put into place and all hardware and software was tested on the Computer Centre, the moment had arrived for moving all hardware (controllers, terminals and printers) to the Library. When the hardware was carried into the Library, the Librarian happened to enter as well and he called out at the project leader: "Ugh, now we have to put up with those machines as well!";
- within the Library, AUBID was part of the deputy manager's responsibilities. For years, he had worked in another part of the DUT as computerisation advisor and did not really want the solution that the Computer Centre offered but wished to find his own solution. He was not all that motivation for the project. However, the HSBI had opted for the DOBIS solution (IBM), so no other solution was possible ;
- the management of the departments involved (middle management) was very motivated and cooperated with the project team in a very constructive way. On the 31st of October 1979, the intermediary request system was officially put into use [see photos jp11]. The middle management had invited the photographic service of the Library. The photographer made a beautiful photo report. The management was at that moment absent.

Summary input and motivation of the management

Management Library:	-
Management HSB1:	0
Management Computer Centre:	+
Management departments Library as involved:	+
Project Management:	+

(- = bad, 0 = moderate, + = good)

Management

To manage an online information system properly demands good management organisation (tasks, responsibilities, competencies, procedures, and so on). To that purpose, the necessary actions were undertaken. As an example, a description of the quality aspect "availability". The Computer Centre is a service organisation. Just like in most computer centres, its customers are as a rule rather discerning. As was the Library. Therefore, criticism regarding the availability of the computers of the Computer Centre was often heard. Therefore, in order to avoid any possible problems it is advisable to come to a clear written arrangement about this in a Service Level Agreement. That was done before the intermediary request system was taken into production. Measurable arrangements were made. The system was allowed to be not available for no more than maximally twice a week, for no more than 10 minutes and once a month for maximally half a day (at least 97% availability). In the first production year, it was still regularly said that the system was insufficiently available. However, adequate test programs showed that the Computer Centre amply fulfilled its obligations.

Picking slip

With every book that is lent, a so-called picking slip with all sorts of data is supplied such as book number and ultimate date of return. The picking slip is also used to provide the reader with some general information. Of course, the project group had no difficulty in designing a picking slip. However, within the framework of involving library staff in the project it was, after consultation with the management, decided to hold a competition for the best picking slip. This resulted in several contributions, including ones from the staff of the "Lending" department, the people that were to use these picking slips every day. After studying the contributions, a picking slip was designed that included several elements of the entries for the competition. In this fashion, the staff was able to see every day that their contribution was valued. The set objective, to involve the library staff as well in this project was achieved.

Methods

- Structured Designing;
- Structured Programming (Nassi Shneiderman, CICS/VS, COBOL);
- Structured Testing;
- Structured Documenting.

Staff

The project manager was dedicated to the project. Among other things he created the functional design. Four staff members were available for doing different jobs.

Size software

20,000 lines of code (estimated).

Author	No	Description	Apply to
		Big Hitters	
BH	01	Poor project management	No
BH	02	Deadlines are unrealistic	No
BH	03	Poor communication	No
BH	04	Incomplete/weak definition requirements	No
BH	05	Insufficient involvement of future users	No
		JS+Others	
JS	RC39	Buyer failure to maintain/enhance system post-implementation	No
JS	PUBRC01	Lack of senior management involvement and commitment	Yes
		Remarks/Observations: It is a matter of the management of the Library being insufficiently involved. However, this is compensated by: • the middle management of the departments in question being strongly involved; • a strong commitment from the project team; • the moral support from the management of the Computer Centre	
PN	10	Technical knowledge is certainly an important skill for project managers to have	Yes
		Remarks/Observations: In view of the technical complexity of the project, the project manager required technical knowledge.	
		Tarek Abdel-Hamid*)	
TAH	01	Software engineering encompasses both the technical aspects of software development as well as the managerial ones {33,47,246}	Yes
TAH	04	There are hundreds of variables that affect software development. Furthermore, these variables are not independent; many of them are related to one another {119,189}. THA constructed *a holistic model* of the software development process	Yes
TAH	08	Two factors significantly influence the initial estimate of the job size: (1) the firm's previous experience; and (2) the general	Yes

		over-all tendency to underestimate the job size {49,223}	
TAH	10	In a multi-project environment, competition for company resources becomes a significant dimension	Yes
TAH	16	A major challenge to managers is to motivate employees. A mechanism for motivation, which is attracting interest in the software engineering field, is "goal setting" {49}	Yes
TAH	18	Newly hired project members pass through an orientation during which they are less than fully productive {75,243}	Yes
TAH	24	The project goals and schedules can play a significant motivational role throughout the life of a software development project	Yes
TAH	33	Progress in a software project is measured by the number of resources consumed, tasks completed, or both	Yes
TAH	35	As a software project develops, project members often realise that they have under-estimated the number of tasks that constitutes the software system being developed {61}	Yes
TAH	36	There is a powerful tendency to focus on the highly visible mainline components of the software, and to underestimate or completely miss the unobtrusive components (e.g., help message processing, error processing, and moving data around {49}	Yes
TAH	44	By using a model of a complex system, more can be learned about internal interactions than would ever be possible through manipulation of the real system	Yes
TAH	50	For many years, estimation of software project development time and cost had been an intuitive process based on experience and analogy {200}	Yes
TAH	66	A project's estimate creates pressures and perceptions that directly influence the decisions people make and the actions they take throughout the project's life cycle	Yes
TAH	76	QA policy does have a significant impact on total project cost	Yes
TAH	77	An integrated approach helps us achieve an overall understanding	Yes
		*) The list is not exhaustive	

Table 5.8.1: Success/failure factors AUBID project

AUBUD	Score
Complies with functionality agreed	Yes
On time	Yes
Within the agreed budget	----*)
*) no specific project budget	

Table 5.8.2: Results AUBID project

Remarks

The project could have run more pleasantly and quicker if the management of the Library had known SUFFIs TAH/16 and TAH/24 and had acted accordingly.

5.9 CASE: VDV (DUT)

5.9.1 Introduction

In the period 1970-1975, I worked as a system designer/project leader at the Laboratorium Voor Verkeerskunde (Transportation Research Laboratory) of the DUT. During that period, I published the articles jp13 and jp14. This paragraph is about jp13.

5.9.2 Outline article VDV (VerkeersDataVerzameling) (Traffic Data Collection) (1975)

Introduction

The Laboratorium Voor Verkeerskunde (Transportation Research Laboratory), which is (in 1975) part of the Faculteit der Civiele Techniek (Department of Civil Engineering), carries out research into the managing of flow of traffic on roads, crossroads and so on. The purpose of this sort of research is to arrive at a better design of various road elements, such as crossroads, filter lanes and exits and so on, by means of determining motoring behaviour on the one hand and the influence of the factors man, vehicle, road and the like on this. Furthermore, this is an attempt to obtain information about management of the flow of traffic. After all, apart from efficient building, the fast increase of the size of investments in traffic control measures also demands that these are optimally used. Traffic flow models describe the behaviour of the combination of man/vehicle in traffic. However, the complexity of human behaviour limits the applicability of these abstract models for solving practical problems. Through delivery of more detailed data on the flow of traffic, the models can be made more in line with reality. Basically, the flow of traffic measurements in connection with this should register everything that could possibly influence the behaviour of vehicles. However, for the time being this is limited to recording of the *vehicle trajectories.* By a vehicle trajectory is meant a time-road-diagram in which is recorded the road ordinate x as function of the time t. The speed is determined by the formula $v=dx/dt$. A time-road diagram often includes the trajectories of all vehicles that are in a road section during a given period. The study of the flow of traffic at micro level requires information about the *driving of the individual vehicles* on a certain stretch of road.

In 1971, it was decided to computerise the registration and processing of the required measured data to a large degree. Although more information can be registered using film, the processing of film cannot be fully automated (in 1971). By making use of road detectors, this ideal can be achieved sooner. After drawing up of an inventory of the possibilities, an electronic, automatic and mobile traffic registration system was selected that complied with the required (high) degree of accuracy.

The wheels of passing vehicles touch detectors that are put on the road (for example coaxial cables) that give off signals. These are amplified and entered in the measuring vehicle as an electrical pulse, after which these signals are converted into detector information which is processed by the mobile computer and written to magnetic tape.

Literature search and contact with forcign colleagues showed that a similar system did not yet exist.

Traffic data processing

The making of vehicle trajectories from detector reports can be split into the solving of three problems, namely:

- problem 1: bow can *axles* be acquired from detector reports?
- problem 2: how can *vehicles* be acquired from axles?
- problem 3: how is it possible to determine *trajectories* from the vehicles?

Axles

Axles are formed per strip of road in a certain line of direction (two detectors that are placed in one single lane in the same line of direction and are used to detect axles passing). In one line of direction, there are two detectors at an in-between distance of one meter. Assuming that each wheel base is over one meter in size, every axle will first touch detector I and next detector 2 and detector 1 will not be activated again by the next axle until detector 2 has been touched. This way, an alternating series of detector reports is created between detector 1 and detector 2, which is known as the *alternation* between the detector reports of two detectors (see jp13).

An axle is obtained by connection of detector 1 on a registered moment in time with the, in the time scale, first moment in time on detector 2. The registering equipment does not always work faultlessly. The altemation can also be disrupted through other causes, resulting in detector reports being missed or by accident being registered (engines that have not been suppressed for example, may produce a surplus in detector reports under certain circumstances). The software system is required to analyse these types of faults and correct these wherever possible.

Vehicles

If a number of axles on one strip of a line of direction are recorded in a computer memory, then it is possible to determine the speeds of the axles. It is also possible to determine the distance between two successive axles (wheelbase) by means of the time lags. An input parametcr of the system is the maximum allowed wheelbase. Suppose that this is for example 7 meters. In the casc that there are any wheelbases that are larger than 7 meters, thc two axles between which the wheelbase is over 7 meters will not be seen as belonging to the same vehicle. It is possible to make a division between the two axles, a so-called vehicle separator. In the case that two vehicles are driving that close together that the distance between the last axle of vehicle 1 and the first axle of vchicle 2 is less than or equal to 7 meters, then the two axles will be unjustly allocated to the same vehicle. For that reason, the concept of *group* was introduced. By a group is meant: a number of axles that based on the maximum allowed wheelbase are provisionally allocated to the same vehicle until it is demonstrated during the forming of the trajectory that one or more axles are part of a different vehicle. The group is then immediately divided into several groups. The concept of vehicle separator is now replaced by *group separator*, being the mark that is put on an axle when it has been ascertained that this axle is not part of the same group as the previous axle. The number of groups that represents more than one vehicle will strongly depend on the flow of traffic in question. During a number of measurements, over 95% of the groups turned out to be just one single vehicle.

Trajectories

Using the so-called VDV system the trajectory data on the lines of direction in the measured section are recorded in a so-called *master file*. This master file contains master groups. A master group includes, for a number of discrete points, the trajectory data of a vehicle that has passed the measuring section. For every vehicle per line of direction, the following is recorded in the master file: time of passage, speed, time interval and number of the strip in the line of direction.

In order to obtain a master file each group that has passed on line of direction 1 has to be searched on the lines of direction 2 up to and including n. First, the groups of line of direction 1 are linked to a strip group on line of direction 2. The result is a master file based on two lines of direction. Next, an attempt is made to *link* the groups in this master file to a strip group that is part of line of direction 3, and so on. The end result is a master file based on n (= number of lines of direction in the measuring section) lines of direction. Within one line of direction, there will usually be more than one lane. It is the question in which order the lanes have to be traced because it is for example necessary to reckon with vehicles changing lanes (especially in slip roads). Using a so-called relation matrix it is possible to state the *priority relation* between the lanes in line of direction j and the lanes on line of direction j+1. By means of a heuristic search method, it is researched on the basis of so-called *connection criteria*, which group on line of direction j+1 has to be connected to a group on line of direction j. Two connection criteria are used. A criterion for a maximum allowed deviation with regard to the wheelbase (the absolute difference between a wheelbase on line of direction j and a *wheelbase* on line of direction j+1 should be smaller than or equal to delta) and a criterion for the maximum allowed deviation (epsilon) with regard to the *speed*. The values for delta and epsilon are parameterised.

In jp13, the abovementioned problems are further discussed and a sizeable number of memorandums are referred to in which the VDV system, using which the problems were solved successfully, is described.

5.9.3 Management aspects

At the time, two study groups were active within the Laboratorium Voor Verkeerskunde, namely:

- the group "research into the traffic situation" under leadership of MAN-TRAFFIC;
- the group "traffic planning research" under leadership of MAN-PLAN.

Both managers reported to Professor ir. J. Volmuller, Professor of Transport Studies.

VDV system

The commission for developing the VDV system came from MAN-TRAFFIC. The main job was development of the application software for changing detector reports into vehicle trajectories. That was my task. In addition, there was an instrument car with a built-in Raytheon 704 mini computer equipped with a tape unit for recording the detector reports.

Realisation of the technical connections for offering the detector signals to the Raytheon computer was the task of the EMP-TECHOLOGY. The management of the Raytheon computer and the tapes with measuring data were the responsibility of EMP-CONTROL. The actual measurements on the traffic routes were carried out by a measuring team under supervision of the MAN-MEASURING. EMP-CONTROL and MAN-MEASURING reported to MAN-TRAFFIC.

Management involvement and commitment

The involvement of the management team was not a problem at all because the VDV system was of major importance for the study into the flow of traffic. The manual procedures, as used up until now were no longer satisfactory. It is just that ... nobody knew how to solve this problem.

Complexity

Developing the application software of the VDV system was one of the hardest and most fascinating projects in my career. Up to now, it was the only system of which I did not know in advance whether it was possible to solve the problem it presented. When it turned out to function successfully, Professor Volmuller said: "The reason for developing this system, was that such a system did not yet exist. The VDV system is unique in the world!"

Back up

Carrying out a measurement is a sizeable task. After thorough preparation, during which amongst other things permission needs to be obtained from various public bodies, it is (in 1975) usually necessary to put cables and detectors in the road at night. One thing and another can be seen in the pictures included in jp13. Therefore, it caused considerable upset when at a given moment, it turned out that the data (detector reports) of an important measurement were no longer available for transport studies because the EMP-CONTROL had overwritten the tape in question with different data by mistake, whilst no copy had been yet made. This meant that the entire measurement had been in vain and had to be done all over again. This was difficult to explain to the public bodies in question.

Methods

- Structured Designing;
- Structured Programming (FORTRAN IV);
- Structured Testing;
- Structured Documenting.

Staff

The project manager was dedicated to the project. He designed and programmed the VDV model. Two staff members were part-time available for doing different jobs.

Size software

6,000 lines of code.

Author	No	Description	Apply to
		Big Hitters	
BH	01	Poor project management	No
BH	02	Deadlines are unrealistic	No
BH	03	Poor communication	No
BH	04	Incomplete/weak definition requirements	No
BH	05	Insufficient involvement of future users	No
		JS+Others	
JS	PUBRC01	Lack of senior management involvement and commitment	No
JS	PUBRC05	Poor risk management and contingency planning	Yes
PN	10	Technical knowledge is certainly an important skill for project managers to have	Yes
		Tarek Abdel-Hamid*)	
TAH	01	Software engineering encompasses both the technical aspects of software development as well as the managerial ones {33,47,246}	Yes
TAH	08	Two factors significantly influence the initial estimate of the job size: (1) the firm's previous experience; and (2) the general over-all tendency to underestimate the job size {49,223}	Yes
TAH	09	System complexity grows as the square of the number of system elements; therefore, experience with a small system cannot account for all the things that will have to be done in a large system. Neither will the Experience Method apply to systems of totally different content {23}	Yes

TAH	17	All programmers are optimists. They always assume that "This time it will surely run" or "I just found the last bug" {57}	Yes
TAH	24	The project goals and schedules can play a significant motivational role throughout the life of a software development project	Yes
TAH	33	Progress in a software project is measured by the number of resources consumed, tasks completed, or both	Yes
TAH	35	As a software project develops, project members often realise that they have under-estimated the number of tasks that constitutes the software system being developed {61}	Yes
TAH	44	By using a model of a complex system, more can be learned about internal interactions than would ever be possible through manipulation of the real system	Yes
TAH	50	For many years, estimation of software project development time and cost had been an intuitive process based on experience and analogy {200}	Yes
TAH	76	QA policy does have a significant impact on total project cost	Yes
TAH	77	An integrated approach helps us achieve an overall understanding	Yes
		*) The list is not exhaustive	

Table 5.9.1: Success/failure factors VDV project

VDV	Score
Complies with functionality agreed	Yes
On time	Yes
Within the agreed budget	----*)
*) no specific project budget	

Table 5.9.2: Results VDV project

5.10 CASE: BIBLIOSYSTEM (DUT)

5.10.1 Introduction

In the period 1970-1975, I worked as a system designer/project leader at the Laboratorium Voor Verkeerskunde (Transportation Research Laboratory) of the DUT. During that period, I published the articles jp13 and jp14. This paragraph is about jp14.

5.10.2 Outline BIBLIOSYSTEM article (1971)

The fast growing flow of publications results in collecting, arranging and studying of literature takes up an increasingly more important place in research. Therefore, the composition of bibliographies and literature lists is part of virtually every research project of some significance. In this, a large flexibility with regard to updating and classifying proves to be of great value. Using old-fashioned card indexes, these activities were that labour-intensive that these seriously hindered any kind of thorough literature studies, especially when these were of any sort of magnitude. For that reason, the Laboratorium Voor Verkeerskunde decided to computerise the compiling of bibliographies to a high degree in 1971. The system to be developed ("BIBLIOSYSTEM") had to meet the following requirements:

- the system needs to be flexible with regard to updating of the reference file (adding, removing and replacing of references);
- it has to be possible to arrange the references in a specific order;
- it needs to be possible to classify the references in a flexible way;
- as many checks as possible on the coded reference have to be carried out by the system in an efficient way;
- it needs to be possible to provide the references with a "label" in order to be able to refer to these in a report;
- the layout needs to be presentable and it has to be possible to adjust this simply to the (reasonable) requirements of the user;
- the system needs to be as much as possible independent from bibliographic agreements with regard to the way in which references are composed;
- apart from the references, the system should also be able to process the accompanying information;
- the programs need to be able to be used by as many interested parties as possible. This requirement is important with regard to the choice of programming language.

BIBLIOSYSTEM is developed in programming language FORTRAN IV. Over the period 1971-1984, this was used by many groupings within the DUT. In 1984, this was replaced by the GIRAF system [jp9,jp10]. BIBLIOSYSTEM has not just been used for making bibliographies; it has been used as a general system for "automated card-index boxes". The software system has also won its spurs outside the DUT.

5.10.3 Management aspects

At the time (1971), two study groups were active within the Laboratorium Voor Verkeerskunde, namely:

- the group "research into the traffic situation" under leadership of MAN-TRAFFIC;
- the group "traffic planning research" under leadership of MAN-PLAN.

Both managers reported to Professor ir. J. Volmuller, Professor of Transport Studies.

Tests

The commission for development of BIBLIOSYSTEM came from MAN-PLAN, a passionate traffic planning researcher. An interesting detail was that MAN-PLAN said that he also wished to gain programming experience. It was agreed that he was to program one of the programs of BIBLIOYSTEEM under condition that he would observe my method and quality requirements. So he had to develop an adequate programme schedule, then encode this in FORTRAN IV and next test it systematically. MAN-PLAN was allowed to choose the program for himself and did not go for the easiest option. After the program in question had been written it turned out that, for testing it systematically, 100 tests had to be carried out. After having carried out the first twenty tests and not having found any errors in the last few tests, MAN-PLAN thought that enough was enough. "A deal is a deal" was my reaction. Next, MAN-PLAN did carry out all one hundred tests and during the very last test an error did show up. Over the following years, I never needed to convince MAN-PLAN whenever I said that I still had to carry out a few (documented) tests.
MAN-PLAN's attitude was to publish results and to share knowledge. Therefore, he thought it a good idea to publish an article on BIBLIOSYSTEM. After publication of an article in the Dutch magazine "Informatie" in October 1971 [jp14], the Transportation Research Laboratory received more than 100 requests for the system and program description. A number of authorities did in fact decide to put BIBLIOSYSTEM into operation.

Methods

- Structured Designing;
- Structured Programming (FORTRAN IV);
- Structured Testing;
- Structured Documenting.

Staff

The project manager was dedicated to the project. He designed and programmed the BIBLIOSYSTEM information system. One staff member was part-time available for doing different jobs.

Size software

2,000 lines of code.

Author	No	Description	Apply to
		Big Hitters	
BH	01	Poor project management	No
BH	02	Deadlines are unrealistic	No
BH	03	Poor communication	No
BH	04	Incomplete/weak definition requirements	No
BH	05	Insufficient involvement of future users	No
		JS+Others	
JS	PUBRC01	Lack of senior management involvement and commitment	No
PN	10	Technical knowledge is certainly an important skill for project managers to have	Yes
		Tarek Abdel-Hamid*)	
TAH	01	Software engineering encompasses both the technical aspects of software development as well as the managerial ones {33,47,246}	Yes
TAH	08	Two factors significantly influence the initial estimate of the job size: (1) the firm's previous experience; and (2) the general over-all tendency to underestimate the job size {49,223}	Yes
TAH	24	Tbe project goals and scbedules can play a significant motivational role throughout the life of a software development project	Yes
TAH	33	Progress in a software project is measured by the number of resources consumed, tasks completed, or both	Yes
TAH	50	For many years, estimation of software project development time and cost bad been an intuitive process based on experience and analogy {200}	Yes
TAH	76	QA policy does have a significant impact on total project cost	Yes
		*) Tbe list is not exhaustive	

Table 5.10.1: Success/failure factors BIBLIOSYSTEM project

BIBLIOSYSTEM	Score
Complies with functionality agreed	Yes
On time	Yes
Within the agreed budget	----*)
*) no specific project budget	

Table 5.10.2: Results BIBLIOSYSTEM project

5.11 PROJECT AUDITS

5.11.1 Case: Multihouse versus Nutsbedrijven (1997)*)

5.11.1.1 Introduction

In the nineties, Multihouse Automatisering B.V. (also known as MHA) developed a substantial information system by order of the SamenWerkingsVerband NUMIS-2000 (for short SWV).
The SWV was a collaboration between a number of utility companies. The information system had the name NUMIS-2000. In 1991, the parties started this ambitious project. By the end of 1995, the project had run aground. When in 1996, the plan "getting NUMIS-2000 afloat" also was unsuccessful, SWV decided to dissolve the cooperation with Multihouse out of court. The computerisation firm from Gouda tried to prevent this by means of summary proceedings. The judge decided that the SWV had to participate in the winding up of the project test. A further appeal followed in which the judge called in third party experts. Both parties in the case were invited to provide a third party expert. Candidates had to be acceptable to the opposition. After a few candidates were dropped because they were considered to be biased, prof.dr.ir. M. Looijen was proposed by MHA and prof.dr.ir. G.C. Nielen by SWV. Both candidates were appointed as third party expert by the judge and were given the task to suggest a third third party expert together. That was ir. A.J. van Dijk RE. The "Commissie Derde-Deskundigen (Commission Third Party Experts) (for short CDD)" started work towards the end of November 1996. Within a few weeks, an interim report was produced. Early January 1997, a hearing took place at the Court of Justice in Amsterdam. On that occasion, the interim report was also discussed. Both parties interpreted the report to their own advantage, after which the judge asked the representative of the CDD to explain the report. The CDD had a tight schedule, with hearings and wished to listen to both sides, and would deliver its report on the 5th of February 1997. The judge intended to give a verdict on the 7th of February 1997. The judge suggested both parties to wait until the CCD had delivered its final report. SWV was fine with that: "We consider it a good thing to wait for the CCD's final conclusion". SWV mainly wanted to look at matters regarding content. However, MHA asked the judge not to wait but to give his verdict. MHA wished to win the case based on legal arguments.
On the 4th of February 1997, the CDD delivered its report [Van Dijk et al. 1997]. Apart from answering the question as posed, the CDD had also formulated an (unsolicited) advice to parties with the objective to: "Make a substantial contribution to solving the problems that have risen and to supply parties with an opportunity to a (re)new(ed) start". MHA's management agreed with the advice within half an hour but a few days later the judge ruled against MHA. Through basing the proceeding only on legal arguments, MHA had been digging their own grave. MHA's attitude and the outcome of the case were apparently all the more reason for SWV to punish MHA mercilessly. MHA received a claim for

173 million Dutch guilders [Volkskrant 1997]. The sale of MHA, for next to nothing, to Sanderink Beheer (these days known as: Centric) saved MHA from bankruptcy.

5.11.1.2 A few points from the CDD report

- MHA and SWV started a cycle 5 years ago, without either of the parties realising the degree of complexity;
- before the start of the project, MHA and SWV had not performed a risk analysis. If they have done so, then it would have been possible to determine right at the start that development of a large software package for a collection of companies involves large risks. These risks have to be minimised by means of counter measures. From the eighties, there are many examples of similar projects that proved to be a fiasco. There was also literature available on failure factors with regard to developing information systems. Professor Oonincx for example, published as early as 1982 his book 'Why are information systems still failing?' A professional development organisation should certainly know about this. But an assertive customer should also know this;
- immediate involvement of the top management of MHA and SWV in the project was insufficient. Not until the problems had become too large to handle did the management become involved. Moreover, both in general management as well as project management changes took place;
- if SWV had employed EDP-auditors for assessment of propositions made by MHA, it would have meant avoiding of many problems at the start of the project and would in principle have been more economic than using EDP-auditors as soon as things went wrong: in brief, SWV has been leaning too much on MHA, a proactive attitude of SWV would have been desirable;
- because of the long period of realisation, the ambient factors (laws, mergers, etc) changed; this resulted in changes to the functional specifications;
- NUMIS-2000 is a monolithically designed system. Building such a system has a non-linear relation. A monolithic system of for example 6,000 function points is 9 times more difficult to realise than a system of 2,000 function points (quadratic relation). This quality was unknown to both parties at the start of NUMIS-2000 and also remained unnoticed during the building.

The investigations of the CDD were also complex. A large amount of documentation was available but it was difficult to find out what was agreed on exactly, especially at the start of the project. Furthermore, several summary proceedings were simultaneously ongoing between parties and the CDD had to be very meticulous and careful in connection with the large importance attached to the case.

5.11.1.3 Methods

Waterfall

*) A large part of the information in this case was published by journalists in newspapers [Volkskrant 1997, Westerveld 1997].

Author	No	Description	Apply to
		Big Hitters	
BH	01	Poor project management	Yes
BH	02	Deadlines are unrealistic	Yes
BH	03	Poor communication	Yes
BH	04	Incomplete/weak definition requirements	Yes
BH	05	Insufficient involvement of future users	-
		JS+Others*)	
JO	13	The development process takes too long and the costs are too high	Yes
JS	PUBRC01	Lack of senior management involvement and commitment	Yes
JS	RC02	Buyer failure to define clear project objectives, anticipated benefits and success criteria	Yes
JS	RC09	Failure to achieve an open, robust and equitable buyer-vendor relationship	Yes
NB	05	Problems escalated too late	Yes
KY	03	Inadequate project risk analysis	Yes
PN	10	Technical knowledge is certainly an important skill for project managers to have	Yes
		Tarek Abdel-Hamid*)	
TAH	09	System complexity grows as the square of the number of system elements; therefore, experience with a small system cannot account for all the things that will have to be done in a large system. Neither will the Experience Method apply to systems of totally different content {23}	Yes
		Remarks/Observations: In the CDD report, page 4 it says: "The project group has a task to perform. This concerns a large number of activities. Carrying out these activities does require efforts. There is a relation between efforts and results. If a project is simple, or of short duration, then there is often a linear relation between the efforts and results (project category 1). If a project is not that simple, or the project becomes less simple as time goes by, for example because specifications regularly change or because the project is taking (too) long, then the relation between the efforts and results turns out to be no longer linear but more or less quadratic. In that case, the necessary efforts for reaching a result will considerably increase (project category 2). For some projects, it applies that these either take that long and/or become that complex that the relation between efforts and results even becomes exponential. In other words: a lot of efforts for small/limited results (project category 3). For projects in this category, it may be justified to spare no trouble or expense for achieving the desired end result. However, for most projects that end up in project category 3, it goes that these ended up in this category unintentionally. With regard to these projects, only one single conclusion may be drawn: stop immediately.	

		Although a similar classification can also be made for social projects and it therefore may be presumed to be common knowledge, IT practice does regularly prove that (project) managers are not sufficiently or not at all aware of these non-linear relations. The result being that certain projects demand too much effort and deliver too few results and are either adjusted or cancelled far too late in time".	
TAH	12	The longer an error goes undetected, the more extensive the necessary rework and the greater the cost	Yes
TAH	14	The relationship between cost and system size is not linear. In fact, cost increases approximately exponentially as size increases {188}	Yes
TAH	31	We assume that undetected errors will become either "Active errors" (errors that produces more errors) or "Passive errors". Because design specs are the blueprints of the system's code, any errors in design will get translated into coding errors	Yes
TAH	42	Using the system dynamics model as an experimentation vehicle can provide many benefits to the software engineering community	Yes
TAH	44	By using a model of a complex system, more can be learned about internal interactions than would ever be possible through manipulation of the real system	Yes
TAH	52	By imposing different estimates on a software project we create different projects. Research clearly indicates that the decisions that people make in projects and the actions they take are significantly influenced by the pressures and perceptions produced by the project's schedule {57,109,125,224,236}	Yes
TAH	77	An integrated approach helps us achieve an overall understanding	Yes
		*) The list is not exhaustive. If all/most Big Hitters are set to Yes, then the number of other failure factors are usually countless.	

Table 5.11.1.1: Success/failure factors NUMIS-2000 project

NUMIS-2000*	Score
Complies with functionality agreed	No
On time	No
Within the agreed budget	No
*) on 4 February 1997	

Table 5.11.1.2: Results NUIMIS-2000 project

Remarks

The TAH SUFFIs provide more understanding. TAH/09 is a very important SUFFI. However, this may also be viewed as part of KY/03: *Inadequate project risk analysis*.

It may be presumed that if the (project) management had been aware of the SUFFIs TAH/09 and TAH/77, the results of the project in question would have been better.

5.11.2 Case: SYSA (GOVERN) (2004)*)

5.11.2.1 Introduction

Within the Dienst ABC (ABC Service) (DABC) of public body GOVERN, Unit A (UA) and Unit B (UB) function amongst others. For its activities, UA makes use of the PROGA software system for registration of report data. For its activities with regard to maintenance of equipment, UB makes use of the PROGB software system.

In August 2001, the report "SAP R/3 for replacement of PROGA and PROGB" was issued. The reasons were that the applications were obsolete and did not work integrated.
Conclusion in the report: "The functionality as demanded by the DABC can be housed with the already available SAP R/3 modules in GOVERN. A start can be made with the implementation of the current PROGA functionality in the available SAP modules, to be followed by the PROGB ones. Next, the functionality can be expanded within the available modules. For the time being, it is advisable to use the available modules within SAP R/3". In January 2003, the Plan van Aanpak (Plan of Approach) was delivered for system A (SYSA). The SYSA project was given the assignment:
"Build and realise within SAP R/3 the current functionality of PROGA and PROGB, complying with the processes as described in the document 'Dienst ABC Inrichtingsplan' (ABC Service Design plan) and the document 'end report SAP R/3 for replacement of PROGA and PROGB'. This involves 1-on-1 transfer of the functionality".

Stopping with SAP/SYSA?

In the note of February 2004, the DABC draws the following conclusion:

- SAP/SYSA can only be made and kept workable by means of hefty investments in money and manpower. Sufficient insight was gained into the working of SAP/SYSA for drawing this conclusion;
- the implementation of SAP in the current structure of UB does involve safety risks. A radical change of the UB structure and the processes because of the introduction of a supporting computer system is not desirable;
- although a lot of money has been invested in SAP/SYSA already, it is recommended to nevertheless stop the project in order to avoid further escalation of expenditure.

5.11.2.2 Investigation

With reference to the advice to stop SAP/SYSA, the head of ICT-Operations (HOPER) of the Dienst Informatievoorziening (Information provision Service) (DINF) has proposed to have a brief investigation (audit) done by an independent researcher/auditor. Aart van Dijk (AvD) was asked to carry out this investigation. He accepted the commission under the express condition that he would be able to carry out an independent/impartial investigation.

The first activity was a round of talks. AvD has conducted conversations: asked questions but first and foremost listened. As a result of the talks, an atmospheric description was made and discussed with the head of DABC. The latter stated that the DABC only needs the judgment of an independent expert, who answers a few questions and links this with advice. The DABC wanted answers to the following questions:

1. Are the (technical) preparations of sufficient quality for this to be a solid basis for continuing this SAP project?
2. Is it possible to amend SAP/SYSA such that it does indeed deliver a system that supports the processes of the DABC?
   a. Does the SAP/SYSA system as delivered meet the requirements that were agreed?
   b. Which adjustments are needed for increasing the user-friendliness?
3. Which preconditions are yet to be fulfilled (for instance taking care of the functional management, restoring of trust)?
4. Which investments still have to be made (by DABC and DINF) in people and means for properly rounding off the project?

5.11.2.3 Conclusions and recommendations

Conclusions

Some conclusions were [Van Dijk 2004-1]:

- both parties are responsible for the quality, progress and depth of the activities that were carried out (see Chapter 1);
- both parties have underestimated the SAP/SYSA project;
- both parties (head of DBAC and manager applications (MANAPP) of DINF) have given too little attention to the management of the project;
- an information system consists of five components: hardware, software, data ware, org ware and human ware. Too little attention was given to the last two components;
- wrongly, the following does not prevail within the project "a deal is a deal". People have the tendency to put agreements up for discussion or to ignore these. The probable cause was that some people preferred another solution;
- the conversations have proved that the choice of SAP by DABC was not a spontaneous one but did come about after some insistence of DINF's MANAPP. During the project, this is from time to

time noticeable in some DABC staff, who in that case ask themselves (again) whether the correct choice was made;

- the atmosphere between the parties displays a "mutual motion of no confidence";
- DINF is prepared, more than before, to reckon with the interests/wishes of the DABC staff by delivering (more) made-to-measure SAP. In doing so, a concession is made with regard to the management component;
- the Plan van Aanpak (Plan of Approach) that is included in appendix 1, indicates how the functionality that the DABC reported as missing, can be realised at a very late stage. This Plan van Aanpak is provided with comments by DABC and DINF;
- PROGA was not transferred 1-on-1. It is a matter of extra functionality. Instead of a fragmented toggle system, there is now an integrated system in SAP;
- for some users of DABC the new system has become too difficult. On the one hand because SAP/SYSA/PROGA uses many more display images and on the other hand because SAP/SYSA/PROGA is used very irregularly by individual employees;
- to enable a successful completion of the project, extra financial means are necessary. The extra amount of money that is needed for completing SYSA in accordance with the formulated principles is at least € 250,000. - and is higher than the amount as planned for realisation of the entire project (€ 220,000.-).

Recommendations

- starting point should be/remains that, no matter how, the DABC gets an adequate information system for replacement of the products PROGA and PROGB;
- in determining which solution approach will be used, do not just assume the already made investments but do especially take your starting point from the investments that still need to be made for achieving the set goal;
- when determining the solution approach do not just look at the DABC's interests, although these interests do weigh heavy, but also look at the interests of GOVERN, such as integration and standardisation;
- when the solution approach does remain SAP/SYSA, do make some tailor-made adjustments to SAP. In doing so, you should think of some exits and a few DABC process-dependent display images ("ancillary display images") which will help achieve the necessary user-friendliness;
- appoint an independent project leader who is given competencies, such as escalate if necessary up to the highest level of management, when agreements are not honoured.

PROGA

- in the first instance, select solution approach 1;
- have the employees continue to check the registration forms and if necessary add to these but do not allow them to enter data in SAP/SYSA/PROGA;
- put some employees of the DABC in charge of entering the data of the registration forms. They can be given thorough training in SAP/SYSA/PROGA. They will keep this knowledge up-to-date by using it on a daily basis. It should be reviewed to see whether other staff could be or wish to be involved;
- be aware of the fact that a new system sometimes also involves having to adjust the administratively organised procedures;
- properly test the new system and do not change over to the new system until the employees in question are entirely confident using the new system;
- if after delivery and serious testing of SAP/SYSA/PROGA there is not a properly working system available, then go for solution approach 3 or 6.

PROGB

- in the first instance, select solution approach 1;
- investigate the possibilities of SAP module XYZ and the other possibilities that are included in the Plan van Aanpak from appendix 1;
- decide to continue with solution approach 1 or if the investigation does not give any satisfactory results then opt for solution approach 3 or 5.

5.11.2.4 Reaction

During the first talk with the head of DABC, he held a long tirade about DINF. That is where all the problems were. It was up to me, he said, to make it clear to them once again. The DABC was not to blame. I pointed out to him that as a registered auditor I was going to carry out an independent/impartial investigation based on the rules of conduct and professional standards of the NOREA. During the discussion of the concept report with all parties involved, there was a distinct lack of introspection in the bead of DABC. The report of the IT Auditor (AvD) was no good, the advice of the IT Audior (AvD) was no good and the IT auditor (AvD) was no good. He even openly threw doubt upon the IT auditor's integrity. I rounded off the report without making concessions to my findings. Paraphrasing the well-known Dutch Professor Strikwerda: "When you do not have a straight back, you should not become/remain an auditor but you would do better looking for another profession". The next day, HOPER informed the head of DABC that his behaviour was not just incorrect but that in view of the status of the IT auditor (working in compliance with the rules of conduct and professional standards of

the NOREA) his behaviour could very well have legal consequences. The very same day, the head of DABC offered me his, in my view not sincere, apologies.
It is interesting that six months on my advice with regard to PROGA was taken on board fully and an investigation into solution approach 1 with regard to PROGB as recommended by me was underway.

5.11.2.5 Methods

Waterfall

*) Some names in this case have been invented. The case itself is truthful.

Author	No	Description	Apply to
		Big Hitters	
BH	01	Poor project management	Yes
		Remarks/Observations: "DABC itself wished to lead the project. The Head of UB is the project manager"	
BH	02	Deadlines are unrealistic	Yes
BH	03	Poor communication	Yes
BH	04	Incomplete/weak definition requirements	Yes
		Remarks/Observations: June 2003: Confusion: "What are in fact the requirements?"	
BH	05	Insufficient involvement of future users	Yes
		Remarks/Observations: "DABC often does not honour agreements, they have too little time and show little involvement"	
		JS+Others*)	
JO	13	The development process takes too long and the costs are too high	Yes
JS	PUBRC01	Lack of senior management involvement and commitment	Yes
		Remarks/Observations: At the start, there was involvement but that soon disappeared. "DABC shows too little involvement"	
JS	RC02	Buyer failure to define clear project objectives, anticipated benefits and success criteria	Yes
		Remarks/Observations: "It is no good to leave things as they are. However, proposals are not taken seriously. We can't really do any good anymore"	
JS	RC09	Failure to achieve an open, robust and equitable buyer-vendor relationship	Yes
		Remarks/Observations: "There is not trust". "They don't want to". "Too little feeling for achieving a result together". There was a lot of turnover amongst external employees because they were fed up with the project.	
NB	05	Problems escalated too late	Yes
		Remarks/Observations: "It has recently been reported that functionality is lacking but that has never happened in the steering committee"	
KY	03	Inadequate project risk analysis	Yes

PN	10	Technical knowledge is certainly an important skill for project managers to have	Yes
		Tarek Abdel-Hamid*)	
TAH	01	Software engineering encompasses both the technical aspects of software development as well as the managerial ones {33,47,246}	Yes
TAH	02	Poor management can increase software costs more rapidly than any other factor {261}	Yes
TAH	05	The behaviour of an individual subsystem in isolation may be different from its behaviour when it interacts with other subsystems {68}	Yes
TAH	09	System complexity grows as the square of the number of system elements; therefore, experience with a small system cannot account for all the things that will have to be done in a large system. Neither will the Experience Method apply to systems of totally different content {23}	Yes
TAH	13	Inadequate planning is the primary reason for loss of control on many computer programming projects {208}	Yes
TAH	14	The relationship between cost and system size is not linear. In fact, cost increases approximately exponentially as size increases {188}	Yes
TAH	22	Problems of communication and motivation are responsible for inadequacies in process and for consequent losses in productivity {240}	Yes
TAH	24	The project goals and schedules can play a significant motivational role throughout the life of a software development project	Yes
TAH	35	As a software project develops, project members often realise that they have under-estimated the number of tasks that constitutes the software system being developed {61}	Yes
TAH	42	Using the system dynamics model as an experimentation vehicle can provide many benefits to the software engineering community	Yes
TAH	52	By imposing different estimates on a software project we create different projects. Research clearly indicates that the decisions that people make in projects and the actions they take are significantly influenced by the pressures and perceptions produced by the project's schedule {57,109,125,224,236}	Yes
TAH	77	An integrated approach helps us achieve an overall understanding	Yes
		*) The list is not exhaustive. If all/most Big Hitters are set to Yes, then the number of other failure factors are usually countless.	

Table 5.11.2.1: Success/failure factors SYSA project

SYSA*	**Score**
Complies with functionality agreed	No
On time	No
Within the agreed budget	No
*) in February 2004	

Table 5.11.2.2: Results SYSA project

Remarks

The TAH SUFFIs provide more understanding. TAH/09 is a very important SUFFI. However, this may also be viewed as part of KY/03: *Inadequate project risk analysis*. It may be presumed that if the (project) management had been aware of the TAH SUFFIs mentioned in table 5.11.2.1, the results of the project in question would have been better.

5.11.3 Case: ACCINT (PUBLIC) (2004)*)

5.11.3.1 Introduction

The ACCINT project has the following objectives (source: PID Implementation ACCINT):

- introduction of a set code of conduct 'dealing with information';
- introduction of the ACCINT internet *functionality* (both the e-mail as well as the browse functionality) in a controlled and as safely as possible way for obvious users;
- phasing out of the current internet *facilities* (predominantly stand alone pc's) simultaneously with the above;
- supplying of an additional *facility* for those that are able to substantiate that the ACCINT functionality will not suffice them for executing their function.

The "Rollout of the Browse functionality" of the ACCINT project does not go entirely as planned. ACCINT's client, Mr Xyz who is on the board of directors, in the meantime issued commitments to the departments, which have not been fulfilled.

The CIO of PUBLIC has commissioned AvD to carry out an audit/investigation into ACCINT with the objective to provide an answer to the question "What went wrong?".

5.11.3.2 Main conclusions, recommendations and points open to improvement

Conclusions [Van Dijk 2004-2]

- the project manager (PM) of ACCINT has asked the Head of ICT Operations in August 2003 to 'take (have) matters (taken) in hand in the (technical) discussions because the ACCINT project only looks at the issues in a functional way'. In my opinion, that is an understandable attitude of the PM. However, ICT Operations has hardly enough capacity for Technical Project Managers (TPM) and was initially not in a position to provide TPM. ICT Operations did however, at the request of the PM supply a coordinator;
- in the starting phase of ACCINT too little specific technical knowledge was brought into the project team, also in talks with suppliers. This was one of the reasons that existing knowledge and expertise as present in ICT Operations was used insufficiently and time was wasted. Furthermore, this also led to the problem with regard to the finances of the network components and was discovered too late that the solution as given by the supplier SUPPL did not include the possibility to register the activities of individual users;

- no documentation has been found about the way in which the choice for SUPPL was made (such as requirements in writing, reports of conversations and motivation for selection of SUPPL). The department of Financiële Zaken (Financial Affairs) is of the opinion that the acquisition process does not comply with the going standards;
- the negotiations with the suppliers took place on the basis of general requirements. During the course of the project, the requirements were changed, such as the addition of the mail facility. There is no up to date overview/document of the Program of Demands. ICT-Operations are not fully informed of all sorts of specific requirements that came forward from the inventory. This is experienced as being limiting because the end user sometimes proves to have expectations that cannot be realised. The PM thinks that this knowledge is not essential because ICT-Operations only supplies the standard facilities and the special cases will not be worked on until a later stage;
- within ICT-Operations, too little expertise is available at project management level. Some staff will be given the role of project manager while they are not trained to do this and also have different types of tasks;
- within ICT-Operations, there is little thematic work going on and hardly any priorities are allocated. Employees often work on several tasks at the same time but do not always have a view of the mutual priority of the tasks. This sometimes causes friction with project managers/clients who approach employees of ICT-Operations directly;
- departments of PUBLIC offer work to ICT-Operations, but often do not state what priorities this work has as compared to other work they offer (when everything has priority then nothing has priority anymore!). The PM has indicated that the management of ICT-Operations has been expressly confronted with the priority of ACCINT through interventions of the strategic management. The Head of ICT-Operations when asked about this reacted as follows: 'You have to realise that ACCINT is just one of the many projects that ICT-Operations are working on. Currently, there are between fifteen and twenty large projects in progress that have the same high priority as ACCINT';
- because of the considerably greater interest for the browse functionality, the sub project 'Rollout Browse functionality' got a different dimension in its execution;
- the head of ICT-Operations responded flexibly to the financial aspects of this changed scope;
- an change of TPM took place, at the cost of the continuity;
- the rollout at department S required special attention and time because of a number of technical complications involving third parties;
- the SUPPL company did apologise for that part of the delay of ACCINT that can he attributed to SUPPL;

- sending of confidential or privacy-sensitive data is on principle not permitted but has to be supported in relation with any exceptional situations.

Recommendations

- be aware that ACCINT is a project with a great relevance to PUBLIC. The volume of sub project ICT is perhaps limited but should not be underestimated because of the (challenging) technical complexities;
- projects are always urgent, so do forget the word urgent. Ensure the correct priority (in fact, the mutual priority at demand side level should be arranged);
- make an overview of tasks, responsibilities and competencies. Also record who supplies what information and when. Do not create a voluminous plan, make it a short and clear overview (in fact, this should be arranged in cooperation with the demand side);
- have ICT-Operations make a (updated) detailed overview of which activities have to be executed for installing an internet PC and include an estimate time;
- have ICT-Operations make a plan on the basis of the previous point for the rollout based on recent documentation supplied by the PM;
- have ICT-Operations make a (updated) detailed overview of the activities that have to be executed at department S for an internet PC;
- with regard to ACCINT only work on agreed and recorded tasks;
- make sure to have an up-to-date overview with regard to the Program of Demands. Have functional changes run via Change Management for impact analysis, and so on, and so forth;
- let bygones be bygones and do realise that it is 'the tone that makes the music';
- do realise that a relationship with supplier SUPPL is a relationship that will have to last for a long time and invest in that relationship;
- make sure that the PM has one single contact within ICT-Operations, thus avoiding him having to shop around within ICT-Operations.

Points for improvement

- set up a projects organisation within ICT-operations with a program manager, project office and TPM-ers (action: ICT-Operations);
- within ICT-Operations make sure that employees know which (mutual) priorities their activities have and have them report frequently on the progress (action: ICT-Operations);

- within PUBLIC, set up a Stuurgroep Bestuurlijke Informatievoorziening (Steering Committee Administrative Information Provision) (SBI) that amongst other things defines projects and determines priorities.

5.11.3.3 Reactions

ICT-Operations

Although ICT-Operations has received the necessary criticism in the report, the head of the department did agree with the conclusions.

Project Manager

The PM was young and inexperienced. He was supported by the client. Of course, that is fine but just as in the Telephony project in section 5.5, in my opinion the client had total confidence in the PM without remaining critical. That can be dangerous and as happened in this case; it can result in a "troubled project".
The PM took great offence that the CIO had ordered this audit. He considered it a vote of no confidence (in my opinion, a professional PM will order the odd audit himself occasionally). According to the PM the audit report as well as the auditor were no good. After a few collisions with the CIO, the PM was ill at home for a long period. An appointed external PM stated, after he had been shown the ropes, that the audit report did give a truthful image of reality.

5.11.3.4 Methods

- Prince 2 project management method;
- no software development.

*) Some names in this case have been invented. The case itself is truthful.

Author	No	Description	Apply to
		Big Hitters	
BH	01	Poor project management	Yes
BH	02	Deadlines are unrealistic	Yes
BH	03	Poor communication	Yes
BH	04	Incomplete/weak definition requirements	Yes
BH	05	Insufficient involvement of future users	Yes
		Others*)	
JO	13	The development process takes too long and the costs are too high	Yes
JS	PUBRC01	Lack of senior management involvement and commitment	No

		Remarks/Observations: It should be remarked that blind faith of a manager in one single employee without remaining critical is very bazardous and, as happened in this case, may lead to a "troubled project".	
JS	RC02	Buyer failure to define clear project objectives, anticipated benefits and success criteria	Yes
JS	RC09	Failure to achieve an open, robust and equitable buyer-vendor relationship	Yes
NB	05	Problems escalated too late	Yes
KY	03	Inadequate project risk analysis	Yes
PN	10	Technical knowledge is certainly an important skill for project managers to bave	Yes
		Tarek Abdel-Hamid*)	
TAH	02	Poor management can increase software costs more rapidly than any other factor {261}	Yes
TAH	07	People under time pressure don't work better, they just work faster. ... In the struggle to deliver any software at all, the first casualty bas been consideration of the quality of the software delivered {81}	Yes
TAH	10	In a multi-project environment, competition for company resources becomes a significant dimension	Yes
TAH	12	Tbe longer an error goes undetected, the more extensive the necessary rework and the greater the cost	Yes
TAH	13	Inadequate planning is the primary reason for loss of control on many computer programming projects {208}	Yes
TAH	16	A major challenge to managers is to motivate employees. A mechanism for motivation, which is attracting interest in the software engineering field, is "goal setting" {49}	Yes
TAH	22	Problems of communication and motivation are responsible for inadequacies in process and for consequent losses in productivity {240}	Yes
TAH	24	The project goals and schedules can play a significant motivational role throughout the life of a software development project	Yes
TAH	28	Design errors in the early design pbase are generated at a higher rate than are coding errors {165}	Yes
TAH	31	We assume that undetected errors will become either "Active errors" (errors that produces more errors) or "Passive errors". Because design specs are the blueprints of the system's code, any errors in design will get translated into coding errors	Yes
TAH	36	There is a powerful tendency to focus on the highly visible mainline components of the software, and to underestimate or completely miss the unobtrusive components (e.g., belp message processing, error processing, and moving data around {49}	Yes
TAH	42	Using the system dynamics model as an experimentation vehicle can provide many benefits to the software engineering community	Yes
TAH	44	By using a model of a complex system, more can be learned about internal interactions than would ever be possible through manipulation of the real system	Yes
TAH	45	A full-time employee allocates, on the average, 60% of his or her time to productive work on the project	Yes
TAH	49	Project members are not willing to maintain an above-normal work rate indefinitely. Once people start working at a rate above their normal rate, their "Overwork Duration Thresbold" decreases because people enjoy and need their slack time	Yes
TAH	60	The policy of allocating project members half-time to the project results in a cost that is about 22% higher	Yes

TAH	68	When a software development project overruns its schedule there is an apparent cause: the project was poorly estimated	Yes
TAH	77	An integrated approach helps us achieve an overall understanding	Yes
TAH	80	A different schedule creates a different project	Yes
		*) The list is not exhaustive. If all/most Big Hitters are set to Yes, then the number of other failure factors are usually countless.	

Table 5.11.3.1: Success/failure factors ACCINT project

ACCINT*	Score
Complies with functionality agreed	No
On time	No
Within the agreed budget	No
*) in June 2004	

Table 5.11.3.2: Results ACCINT project

Remarks

The audit report that I wrote quotes the following remark [Rijsenbrij & Kouwenhoven 1996]:
"the objective of a project management audit is to obtain an independent judgment about the situation on a project with regard to the order and project management. The reason for carrying out an independent study is that it enables officials that are not immediately involved in the project to indicate from their own experience and background where and why certain things are in trouble. A project management audit is mainly intended to map the present as well as the future. An audit is not suitable (neither intended) to assess people. One bas to remember that 80% of the errors are not incidental but are caused by the working method in an organisation and thus come under the responsibility of the management" [Van Dijk 2004-2].
Tbe conclusions, recommendations and points open to improvement, as established in paragrapb 5.11.3.2, also indicate that this project could bave run considerably better if the working method within PUBLIC bad been more professional.

5.11.4 Case: SOX (FINANCE) (2006)*)

5.11.4.1 Introduction

As part of SOX, FINANCE*) a well-known financial institution in the Netherlands carried out audits/tests on the IT infrastructure in the period mid August until mid December 2006. I was one of the IT auditors and carried out audits on a number of realised IT projects. Fully failed projects did not come up for discussion. This paragraph, first briefly discusses SOX. Afterwards, the results of the project audits will be discussed and I will briefly discuss the Sóxima model that I constructed for the benefit of FINANCE .

5.11.4.2 SOX and SOX 404

In 2002, the Sarbanes-Oxley Act ("SOX") was passed in the USA. This more strict legislation in the field of financial reports should limit the chance of accountancy scandals such as happened at for example at Enron, Worldcom and Ahold. All businesses that are quoted on the American Stock Exchange are obliged to comply with this act.

The act aims at restoring the confidence in the financial markets by setting higher demands to the quality and set-up of the administrative processes and internal control. To that purpose, the CEO and CFO have to provide an assessment about the effectiveness of the internal control system at the end of the financial year. For every periodical report, they are required to have this certified again by the external accountant. Because it is quoted on the New York Stock Exchange, the act is applicable to FINANCE, more precisely from the financial year 2006.

Some characteristics of section 404 of SOX:

SOX 404 sets rules for internal controls and financial reports. The management is obliged to make an explicit annual statement about the reliability of internal controls as performed within the business and bears personal responsibility with regard to this. The CEO (Chief Executive Officer) and the CFO (Chief Financial Officer) have to issue a statement which says that the report is a correct representation of the state of affairs and does not include any substantial contraventions. In addition to his usual task in the field of financial reporting, the auditor (accountant) has to add an explicit statement about his agreement with the statements of the CFO and CEO.

This amounts to the financial annual report having to include a chapter every year, which states that the internal control does evaluate the correctness of the figures as offered. The internal control as performed by businesses concerns the systems as used by businesses for checking whether the reported figures are correct. The objective is to prevent creation of the wrong image based on incorrect or incomplete data, on the basis of which the wrong decisions might be made. Initially, this control is aimed at ensuring that the internal information is correct, thus ensuring that the management makes decisions based on the correct data. Secondly, this also means that the data as presented by a business is verified. In all this, IT takes a special place. IT is not an immediate goal of SOX, that is the internal control, however, within this internal control, IT plays a central role. In the internal control, a framework of internal control is used: COSO**).

SOX sanctions

Should FINANCE not comply with the SOX act, this may amongst other things result in:

- the FINANCE share being crossed off at the NYSE;
- damaged reputation;
- drop in share prices;
- criminal prosecution; up to $ 5,000,000 fine and/or 20 years imprisonment for the CEO and CFO.

What does SOX means concretely for FINANCE

- being in control, nothing new;
- being able to prove this (transparency), that is new;
- CEO and CFO personally sign for this. Basis previously: internal sign off by managers;
- the line (the manager) is therefore responsible for being in control of his/her area and for transparency.

*) Some names in this case have been invented. The case itself is truthful.

**) COSO stands for *Committee Of Sponsoring Organisations of the Treadway Commission.*
One of the most important guidelines is the COSO report "Internal Control-Integrated Framework", which was published in 1994. The COSO report provides a definition of internal control and supplies a few standards that the internal control of a company has to comply with.

5.11.4.3 Results

In general, it applies that "being in control" is nothing new to companies and institutions. However, it is new that some companies have to be able to prove this as part of SOX (transparency). To put it briefly: the adage "tell me" has changed into "show me". This has a major impact on the companies themselves, which was also obvious during these project audits.

As much as possible, the projects were drawn at random from the collection of known (registered) projects.

Findings [Van Dijk 2006]

General

- in projects, a Business Case was used. However, after approval of the project little attention was paid to the Business Case;
- in practice, test plans were drawn up and carried out. It is not formally enforced that tests are always carried out and the results of the tests are not formally filed away. Also, no requirements have been set with regard to tests;
- although some processes were consistently pursued, this was not always demonstrable;
- it cannot be proved that only the tested and authorised software was taken into production;
- Prince2 was used but because of the combining of a few stages, there was no formal enforcement with regard to explicit milestone moments between realisation, testing and implementation;
- no structured post-implementation reviews were carried out;
- some projects required more time than was planned or cost more than was foreseen.

Some findings concerning individual projects

- no formal (written) approval by the management;
- no architecture plan was made;
- the PID does not include CIA-levels (Confidentiality, Integrity, Availability);
- no explicit permission was found for the transition from development phase to test phase;
- approval for the transition to production has not been found;
- no specific acceptance criteria;
- there were no logically separated DTA (Develop, Test, Acceptance) environments for this project;
- after implementation, the architect did not carry out an architecture check.

FINANCE has in the meantime taken steps for being able to comply with the SOX requirements ("Show me").

5.11.4.4 Methods

- Project management method: based on Prince 2;
- System development: depends on the project: waterfall, prototyping

Remark

In order to be able to test all SOX requirements, a large number of detail factors will have to be added to the "Reference model success and failure factors ICT projects" (SUFFI model for short). However, that does not make the model more surveyable.

Author	No	Description	Apply to
		Big Hitters	
BH	01	Poor project management	No
BH	02	Deadlines are unrealistic	No
BH	03	Poor communication	Yes
BH	04	Incomplete/weak definition requirements	No
BH	05	Insufficient involvement of future users	No
		Others*)	
JO	13	The development process takes too long and the costs are too high	Yes
JO	17	Insufficient attention to reliability and controllability	Yes
JS	RC22	Failure to undertake effective project reviews and take decisive action	Yes
KY	08	Poor (written) internal communication (SOX)	Yes
NB	06	In cases of failure there is too little appreciation and attention to the quality of the (written) communication	Yes
PN	06	The use of a Business Case results in a higher degree of satisfaction with the project, whilst the satisfaction with the project is very low when no Business Case is used**)	Yes
PN	14	Evaluating projects is strongly related to the satisfaction with the successful execution of projects	Yes
		Tarek Abdel-Hamid*)	
TAH	01	Software engineering encompasses both the technical aspects of software development as well as the managerial ones {33,47,246}	Yes
TAH	05	The behaviour of an individual subsystem in isolation may be different from its behaviour when it interacts with other subsystems {68}	Yes
TAH	07	People under time pressure don't work better, they just work faster. ... In the struggle to deliver any software at all, the first casualty has been consideration of the quality of the software delivered {81}	Yes
TAH	08	Two factors significantly influence the initial estimate of the job size: (1) the firm's previous experience; and (2) the general over-all tendency to underestimate the job size {49,223}	Yes
TAH	10	In a multi-project environment, competition for company resources becomes a significant dimension	Yes
TAH	11	It is difficult to measure performance in programming {181}	Yes
TAH	12	The longer an error goes undetected, the more extensive the necessary rework and the greater the cost	Yes
TAH	16	A major challenge to managers is to motivate employees. A mechanism for motivation, which is attracting interest in the software engineering field, is "goal setting" {49}	Yes
TAH	18	Newly hired project members pass through an orientation during which they are less than fully productive {75,243}	Yes
TAH	19	The training of newcomers, both technical and social, is usually carried out by the "old-timers". This is costly because "while the old-timer is helping the new employee learn the job, his own productivity on his other work is reduced {55,74,77}	Yes
TAH	21	Under schedule pressures, walk throughs and inspections are usually the greatest casualties, they are not only relaxed but often suspended altogether {94}	Yes

TAH	24	The project goals and schedules can play a significant motivational role throughout the life of a software development project	Yes
TAH	28	Design errors in the early design phase are generated at a higher rate than are coding errors {165}	Yes
TAH	32	The earlier the undetected error is, the more "generations" of errors it will produce, and thus the more costly it will end up being	Yes
TAH	34	Progress, especially in the earlier phases of software development, is measured by the rate of expenditure of resources rather than by some count of accomplishments {81}	Yes
TAH	37	In some cases the management becomes increasingly willing to "pay any price" necessary to avoid overshooting the "Maximum Tolerable Completion Date". In such cases, management is often willing to hire more people	Yes
TAH	42	Using the system dynamics model as an experimentation vehicle can provide many benefits to the software engineering community	Yes
TAH	45	A full-time employee allocates, on the average, 60% of his or her time to productive work on the project	Yes
TAH	50	For many years, estimation of software project development time and cost had been an intuitive process based on experience and analogy {200}	Yes
TAH	52	By imposing different estimates on a software project we create different projects. Research clearly indicates that the decisions that people make in projects and the actions they take are significantly influenced by the pressures and perceptions produced by the project's schedule {57,109,125,224,236}	Yes
TAH	54	When the project is perceived to be ahead of schedule, "Parkinson's Law indicates that people will use the extra time for ... personal activities, catching up on the mail, etc. " {49}. This, of course, means that they become less productive	Yes
TAH	65	A different distribution of estimated effort among a project's phases creates a different project	Yes
TAH	69	After all, software estimation is not yet an exact science. Significantly, it is often impossible in a real life situation to demonstrate that underestimation was *not* in fact the cause	Yes
TAH	77	An integrated approach helps us achieve an overall understanding	Yes
TAH	80	A different schedule creates a different project	Yes
		*) The factors have been found at one or more projects. The list is not exhaustive. **) more tests during the project and after completion of the projects are desired.	

Table 5.11.4.1: Success/failure factors FINANCE projects

FINANCE	Score
Complies with functionality agreed	Yes
On time	Yes/No*)
Within the agreed budget	Yes/No*)
*) subject to project	

Table 5.11.4.2: Results FINANCE projects

Remarks

The TAH SUFFIs provide more understanding. It may be presumed that if the (project) management had been aware of the TAH SUFFIs mentioned in table 5.11.4.1, the results of the project in question would have been better.

5.11.4.5 The Sóxima model

During the audits it turned out that it was not always clear what information had to be gathered within the framework of SOX. Therefore, it is advisable to arrive at a detailed SOX information architecture for FINANCE in order to be able to comply with SOX and for demonstrating that FINANCE is in control. In appendix 3, I have shown this by using the Sóxima model [Van Dijk 2006]. Sóxima stands for: SOX Information Management and Architecture.

5.12 DEFINITION OF THE PROBLEM [Answer to sub question 4]

Definition of the problem, sub question 4:

Which SUFFIs are applicable to what particular project from the portfolio of the author's (AvD) projects?

The results of the study are recorded in:

- table 5.2.1: Success/failure factors POTVIS project;
- table 5.3.1: Success /failure factors Kolibrie project;
- table 5.4.2: Success/failure factors Charging method project;
- table 5.5.1: Success/failure factors Telephony project;
- table 5.6.1: Success /failure factors OKAPI project;
- table 5.7.1: Success /failure factors GIRAF project;
- table 5.8.1: Success/failure factors AUBID project;
- table 5.9.1: Success/failure factors VDV project;
- table 5.10.1: Success/failure factors BIBLIOSYSTEM project;
- table 5.11.1.1: Success/failure factors NUMIS-2000 project;
- table 5.11.2.1: Success/failure factors SYSA project;
- table 5.11.3.1: Success/failure factors ACCINT project;
- table 5.11.4.1: Success/failure factors FINANCE projects.

CHAPTER 6

BIG HITTERS / SUCCESSFUL AND NOT SUCCESSFUL

6.1 INTRODUCTION

In chapter 3 and chapter 4 *"The opinion of others about Software Project Management"* is mapped using the "Reference model success and failure factors in ICT projects" (SUFFI model).
In chapter 5 the question is: *"Which SUFFIs are applicable to what particular project from the portfolio of the author's (AvD) projects?"*. The 9 projects on which external publications in the trade magazines appeared are discussed in sections 5.2 up to and including 5.10. In section 5.11, four project audits came up for discussion.

6.2 CONCLUSIONS

As indicated chapter 5 concerns my observations and experience. The 13 cases as described in chapter 5 provide a good picture of those observations and experiences but also leave room for other observations and experiences that I acquired.

Big Hitters

Table 4.2 of the SUFFI model, includes the SUFFIs that have been named as "Big Hitter" by at least 4 authors (see chapter 4):

- poor project management;
- deadlines are unrealistic;
- poor communication;
- incomplete/weak definition requirements;
- insufficient involvement of future users.

John Smith [2001] is of a different opinion: "It is *unwise* to try to rank the root causes into 'big hitters' and 'the rest'. However, such ranking is definitely of value for an *individual* project as it will inform the prioritisation of 'turnaround' actions."

Results cases	Apply to					Score		
	Big Hitter 1	Big Hitter 2	Big Hitter 3	Big Hitter 4	Big Hitter 5	Funct.	On time	Within Budget
Case 1: POTVIS project (KLPD)	No	No	No	No	No	Yes	Yes	Yes
Case 2: Kolibrie project (KPN Telecom)	No	No	No	No	No	Yes	Yes	Yes
Case 3: Charging method project (GAK)	No	No	No	No	No	Yes	Yes	Yes
Case 4: Telephony project (DUT)	No	No	No	Yes	No	Yes	Yes	No
Case 5: OKAPI project (UoA)	No	No	No	No	No	Yes	Yes	---#
Case 6: GIRAF project (DUT)	No	No	No	No	No	Yes	Yes	---#
Case 7: AUBID project (DUT)	No	No	No	No	No	Yes	Yes	---#
Case 8: VDV project (DUT)	No	No	No	No	No	Yes	Yes	---#
Case 9: BIBLIOSYSTEM project (DUT)	No	No	No	No	No	Yes	Yes	---#
--	-------	-------	-------	-------	-------	-------	------	--------
Case 10: Audit Multihouse	Yes	Yes	Yes	Yes	---+	No	No	No
Case 11: Audit SYSA (GOVERN)	Yes	Yes	Yes	Yes	Yes	No	No	No
Case 12: Audit ACCINT (PUBLIC)	Yes	Yes	Yes	Yes	Yes	No	No	No
Case 13: Audit SOX (FINANCE)	No	No	Yes	No	No	Yes	Y/N*	Y/N*

+) unknown
#) no specific budget available
*) Yes or No, depends on the project

Table 6.1: Big Hitters in relation with the discussed cases

Table 6.1 contains the Big Hitters in relation with the discussed cases. Although the results are based on a very limited spot check, the conclusion may be drawn that the collection Big Hitters within this collection of cases acts discriminating. Where at least four of the five Big Hitters are not applicable, the "score" is positive. Where at least four of the five Big Hitters are applicable, the "score" is negative. This picture also corresponds with my experiences and observations during other projects and audits. When the five Big Hitters lead to a negative score, a large number of other SUFFIs usually play a part.

Lack of senior management involvement and commitment

Another SUFFI also plays a major role, namely:

Lack of senior management involvement and commitment (JS PUBRC01)

Results cases	Lack of senior management involvement and commitment	Remark
Case 1: POTVIS project (KLPD)	No	
Case 2: Kolibrie project (KPN Telecom)	No	

Case 3: Charging method project (GAK)	No	
Case 4: Telephony project (DUT)	No	Involvement and commitment were good. However, it should be said that blind faith of a manager in one single employee without remaining critical is very hazardous and, as happened in this case, may lead to a "troubled project".
Case 5: OKAPI project (UoA)	No	
Case 6: GIRAF project (DUT)	No	
Case 7: AUBID project (DUT)	Yes	It is a matter of the management of the Library being insufficiently involved. However, this is compensated by: • the middle management of the departments in question being strongly involved; • a strong commitment from the project team; • the moral support from the management of the Computer Centre
Case 8: VDV project (DUT)	No	
Case 9: BIBLIOSYSTEM project (DUT)	No	
--	------------	--
Case 10: Audit Multihouse	Yes	
Case 11: Audit SYSA (GOVERN)	Yes	
Case 12: Audit ACCINT (PUBLIC)	No	It should be remarked that blind faith of a manager in one single employee without remaining critical is very hazardous and, as happened in this case, may lead to a "troubled project".
Case 13: Audit SOX (FINANCE)	---	This point has not been researched with regard to the SOX projects.

Table 6.2: Lack of senior management involvement and commitment in relation with the cases

I am inclined to consider this SUFFI as the **sixth Big Hitter** but some prudence is in order:

- in the AUBID (DUT) case there was a "Yes" for this SUFFI but thanks to compensating behaviour of others this had no consequences for the end result (the score);
- in the Telephony (DUT) and ACCINT (PUBLIC) cases, there was a "No" for this SUFFI. Nevertheless, at a given moment in time these projects did end up in the status "Troubled project". Thanks to active measures being taken (albeit at a very late stage), the Telephony project was completed successfully. Therefore, there was no *Lack of senior management involvement and commitment (JS PUBRC01)* in this project. In spite of that, something strange did happen. As indicated in 5.5.1, the plan was to make all (a few dozen) small telephone exchanges on the campus defunct. That almost worked out. Only one single small telephone exchange could not be replaced: the one used by the Board of Directors. In my opinion, this is a bad example set by the DUT top management.

Business Case

A SUFFI concerning the Business Case is PN 06: *The use of a Business Case results in a higher degree of satisfaction with the project, whilst the satisfaction with the project is very low when no Business Case is used.* In some projects, there was mentioning of a Business Case (Kolibrie, POTVIS, FINANCE). Based on my experience and observations I am able to endorse the first part of the SUFFI PN 06. A project management method such as Prince 2 does require a Business Case. At every pbase transition, it needs to be established whether the Business Case is still up-to-date/valid. By that, I do also endorse SUFFI PN 07: *The fact whether the casts are evaluated (during and after the project) and whether the benefits are evaluated (at the end of the project) also praves ta have a fairly large influence on the success.*

During some projects, there was a *spin off:*

BIBLIOSYSTEM

After publication of an article in the Dutcb magazine "Informatie" in October 1971 [jp14], the Transportation Research Laboratory received more than 100 requests for the system and program description. A number of authorities did in fact decide to put BIBLIOSYSTEM into operation.

AUBID

Jp12 describes a method that was developed for identifying problems with printers that do not function or do not function properly. This method (the so-called "WEKKER-task" or "ALARM CLOCK-task" method) was developed within the project. Publication in the magazine "Informatie" in April 1980 resulted in the "WEKKER-task"-method being implemented by many businesses;

GIRAF

Apart from a number of applications within the DUT, GIRAF has also been put into operation at the UoA for amongst other things, the information system for the elections [Van Dijk & Keppel 1986] and at the Rijksuniversiteit Leiden (RUL), which used it to create a temporary provision for the thesaurus system of the Library. After the publications and a presentation at an international congress there was also some interest from outside the Netherlands. A large institution sucb as the Vlaams Economiscb Verbond in Antwerp for example, used GIRAF for a number of years for dealing with its mail. After the publications appeared, a few companies in the Netherlands and Belgium introduced the cassette system as developed within GIRAF for all IDMS applications.

A spin off as mentioned above, results in a higher degree of satisfaction with the project.

Evaluation

The cases 1 up to and including 9 are part of the portfolio of projects and relate to 12 published articles. Such an article is in fact a special type of evaluation. Some articles even include a brief evaluation (Charging method, OKAPI, POTVIS). For some projects, an evaluation report was written in addition to the article (for example POTVIS). In my experience, these articles and other evaluations powerfully support SUFFI PN 14: *Evaluating projects is strongly related to the satisfaction with the successful execution of projects.*

Two phases

With regard to the POTVIS project, it was decided to realise this project in two phases. Phase 1 consisted of a number of crucial activities and the other activities were carried out in phase 2. I have used a similar division into two phases before. The advantage of this approach is that you are able to concentrate on the truly essential items in phase 1 and are able to deliver the results of phase 1 sooner. The danger can be that phase 2 is executed partly or not at all. However, it is up to the customers (user organisations) to make an effort for the activities in phase 2. Renewed calibration and determination of priorities at the start of phase 2 are important. In this, the Business Case can play an important role.

Key users

In this chapter up until now Big Hitter BH05: *"Insufficient involvement of future users"* has been given little attention. Within the projects of the cases 1 up to and including 9, this subject got a lot of attention. A few examples:

- AUBID: Library staff were part of the project team. Users were closely involved in the project;
- OKAPI: the key user was not officially part of the project team but did virtually on a daily basis visit the project team and was also involved in presentations to the management;
- BIBLIOSYSTEM: the client has even cooperated in the realisation of the information system;
- POTVIS: users were closely involved in the realisation of POTVIS. They played a particularly important role in the realisation of the RBAC system. They were able to give their opinion in the published article. That was greatly appreciated.

As far as I am concerned, it needs no further explanation that I consider Big Hitter BH05 very important and that a project on this SUFFI should score the "No" value.

Project Manager

"Poor project management (BH01)" is the most often mentioned Big Hitter.
The audit cases SYSA, ACCINT and Multihouse show that there is "poor project management".

Qualities project manager

In Peter Noordam's et al. study [2007], the qualities of the project manager are discussed. Apart from the methodology and project size, the research team has also looked at the project manager as a success factor. In doing so, they looked at three types of projects, with a diminishing degree of technical input:

- *technology projects*: projects in which the technology is the main factor, for example a new operating system or replacement of hardware;
- *integrated projects*: projects in which the technology plays just as important a part as process aspects, for example the implementation of a new information system in which there is significant impact on the (employees within the) organisation;
- *business transformation projects*: projects in which technology plays a secondary part and changing the organisation is the primary objective of the project, for example reorganisations.

Peter Noordam et al. : "Based on the data and the panel discussion it can be stated that there exists no universal project manager who is able to deal equally well with all three types of projects. Albeit that communicative skills are considered important in integrated projects and business transformation projects, in technical projects it will be particularly experience with similar projects that is an important selection factor. It may also be concluded that technical knowledge is certainly an important skill for project managers to have. It may be so that emotional intelligence and functional knowledge are allocated a lot of weight in present-day theory, in practice it does turn out that especially technical knowledge is considered one of the main qualities for concluding projects successfully."

SUFFI PN 10: *Technical knowledge is certainly an important skill for project managers to have.*
This SUFFI occurs in most cases and is therefore confirmed.

Organisation/culture

In my opinion, there are not just three types of projects that matter but also different types of organisations. It does make a difference whether it concerns a purely commercial fixed price project or whether it is a project in the public sector. In my opinion, a different type of project manager will be needed in the first case as compared to the second case. Peter Noordam et al. : "In our firm belief, 70 percent of all projects should not be allowed to start because insufficient thought was given to the added value of the project for the organisation and the minimal preconditions for making the project successful".

I personally have been able to carry out many projects within Dutch universities. Most projects did not comply with Peter Noordam's views. Nevertheless, it was proven possible to make these projects successes. Should I have held to "do not start until most requirements have been met", then a number of projects would not have started at all and therefore would not have been realised at all. The culture does sometimes require that you simply start and find your way within the organisation. That does demand quite a few qualities from the project manager in question, such as for example tact and perseverance. However, broadly speaking, I do agree with Peter Noordam et al.

Reuse of software

Reuse of software is nothing new but is starting to play an increasingly more important part. In particular within the Service Oriented Architecture, which currently receives a lot of attention, reuse of developed modules is a necessity. A professional manager does not just score (delivery of agreed functionalities, on time, within budget, meeting the quality requirements), but also has something good left (software modules that are reusable) at the end of it all. This is an interesting aspect of evaluation.

6.3 DEFINITION OF THE PROBLEM [Answer to sub question 5]

Definition of the problem, sub question 5:

Is it possible to use the Big Hitters for distinguishing between successful and not successful projects from the portfolio of the author's (AvD) projects?

Table 6.1 contains the Big Hitters in relation with the discussed cases. Although the results are based on a very limited spot check, the conclusion may be drawn that the collection Big Hitters within this collection of cases acts discriminating. Where at least four of the five Big Hitters are not applicable, the "score" is positive. Where at least four of the five Big Hitters are applicable, the "score" is negative. This picture also corresponds with my experiences and observations during other projects and audits. When the five Big Hitters lead to a negative score, a large number of other SUFFIs usually play a part.

CHAPTER 7

SUFFI CHARTS

7.1 INTRODUCTION

In the previous chapters, the **SUFFI model** was constructed. It consists of tables 3.7.11, 3.7.12, 3.8.2 and 4.2. The SUFFI model can be put to use immediately and next further tested and improved. In order to simplify its use even more, the tables have been combined into one single table, the so-called **SUFFI Chart**. This is included in section 7.2.

The SUFFI Chart was created from a number of tables (3.7.11, 3.7.12, 3.8.2 and 4.2). These tables include the research results of various researchers. In order to be able to test new research results more easily against all tables, a **SUFFI Total Chart** was created in which all tables (3.7.1-3.7.10, 3.8.1 and 4.2) were included.

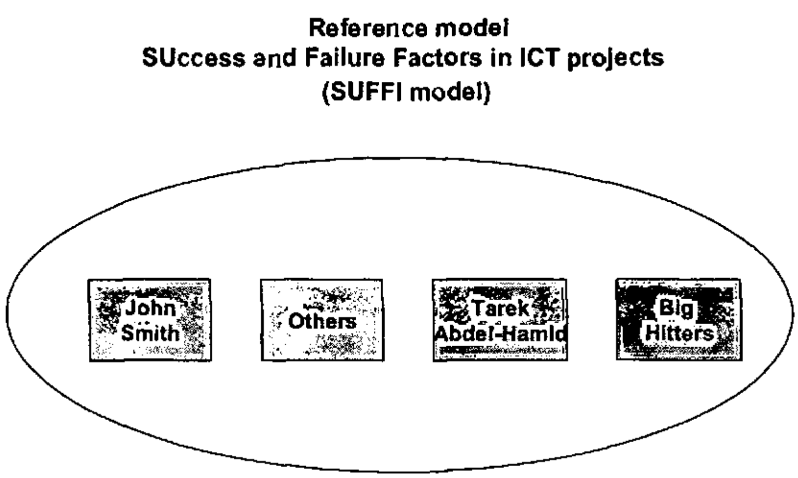


Figure 7.1.1: Reference model SUccess and Failure Factors in ICT projects (SUFFI model)

7.2 SUCCESS AND FAILURE FACTORS IN ICT PROJECTS: SUFFI CHART

Version: 2.0
Date: 1 May 2009
Author: Aart van Dijk

SUccess and Failure Factors in ICT projects (SUFFI Chart)

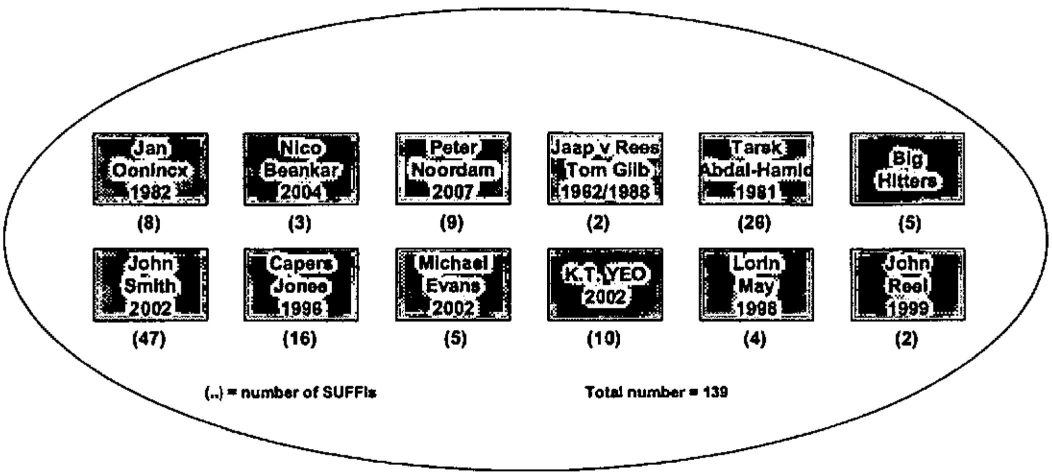


Figure 7.2.1: SUFFI Chart

		Version 2.0 - 1 May 2009
SUccess and Failure Factors in ICT projects: SUFFI Chart		
Author	**No**	**Description**
***Big Hitters*: Success/failure factors that are mentioned the "Big Hitters" (BH)**		
BH	01	Poor project management
BH	02	Deadlines are unrealistic
BH	03	Poor communication
BH	04	Incomplete/weak definition requirements
BH	05	Insufficient involvement of future users
BH	*06*	*Lack of senior management involvement and commitment (see section 9.8)*
BH	*07*	*Lack of professionalism (see section 9.8)*

John Smith, 2002		
Project conception		
JS	RC01	Project based on an unsound premise or an unrealistic business case
JS	RC02	Buyer failure to define clear project objectives, anticipated benefits and success criteria
JS	RC03	Project based on state-of-the-art and immature technology
JS	RC04	Lack of buyer board-level ownership/commitment or competence
JS	RC05	Buyer's funding and/or time-scale expectations unrealistically low
JS	RC06	Buyer failure to break a complex project into phases or smaller projects
Project initiation/mobilisation		
JS	RC07	Vendor setting unrealistic expectations on cost, time-scale or vendor capability
JS	RC08	Buyer failure to define and document requirements (functional and non-functional)
JS	RC09	Failure to achieve an open, robust and equitable buyer-vendor relationship
JS	RC10	Vendor failure to invest enough resources to scope the project to contract
JS	RC11	Buyer lack of sufficient involvement of eventual end-users
JS	RC12	Vendor underestimation of resources (predominantly person-effort) required
JS	RC13	Vendor failure to define project tasks, deliverables and acceptance processes
JS	RC14	Failure to actively manage risks and maintain robust contingency plans
JS	RC15	Poor project planning, management and execution
JS	RC16	Failure to clearly define roles and responsibilities in the contract/subcontracts
JS	RC17	Full-scope, fixed-price contracting (requirements, design and development)
System design		
JS	RC18	Failure to 'freeze' the requirements baseline and apply change control
JS	RC19	Poor choice of technical platform and/or architecture
JS	RC20	Vendor starting a phase prior to completing a previous phase
JS	RC21	Poor choice of design/development method
JS	RC22	Failure to undertake effective project reviews and take decisive action
JS	RC23	Vendor lack/loss of skilled resources
JS	RC24	Poor vendor standards deployment (design, coding, testing, configuration management, etc.)
JS	RC25	Poor vendor requirements traceability (requirements > design > code > test)
JS	RC26	Buyer retention of design authority with right to approve/reject low-level designs
System development		
JS	RC27	Delays causes the project to be overtaken by advances in technology
JS	RC28	Vendor failure to 'freeze' the design (and technical platform) and apply change control
JS	RC29	Inadequate vendor training and supervision of junior staff
JS	RC30	Inadequate vendor review of designs/code/documentation
JS	RC31	Poor vendor management of sub-contractors
JS	RC32	Lack of a formal, 'engineering' approach to integration and testing by vendor
JS	RC33	Insufficient attention paid by vendor to non-functional requirements
System implementation		
JS	RC34	Buyer failure to manage the change implicit in the project (people, processes, technology)
JS	RC35	Inadequate user/systems training
JS	RC36	Catastrophic failure of the system, with no effective contingency arrangement
JS	RC37	Missing a crucial 'go live' date
System operation, benefit delivery, stewardship and disposal		
JS	RC38	Buyer failure to measure actual delivered benefit and take corrective action
JS	RC39	Buyer failure to maintain/enhance system post-implementation
JS	RC40	Changes in the competitive or macro-economic environment
John Smith, 2002 **(UK public sector)**		
JS	PUBRC01	Lack of senior management involvement and commitment
JS	PUBRC02	Failure to focus on key business and end-user needs
JS	PUBRC03	Failure to break complex projects into manageable, separately contracted 'chunks'

JS	PUBRC04	Poor and unimaginative project management
JS	PUBRC05	Poor risk management and contingency planning
JS	PUBRC06	Unclear contracts and poor contract management
JS	PUBRC07	Insufficient focus on user training needs and the design of training interventions
Capers Jones, 1996		
		Large software systems (FP=Function Point)
CJ	01	1.000 FP's: Quality control is a major requirement at this size range
CJ	02	1.000 FP's: With team development (up to 10 staff members), issues of system segmentation and interfaces among components become problematic
CJ	03	10.000 FP's: This size range is often plagued by cost and schedule overruns and by outright cancellations
CJ	04	10.000 FP's: Development teams of 100 or so are common, so communication and interface problems are endemic
CJ	05	10.000 FP's: Configuration control and change management are mandatory for this size plateau
CJ	06	100.000 FP's: Development teams number in the hundreds, often in multiple locations that may even be in different countries. Communication problems are rampant
CJ	07	100.000 FP's: Formal configuration control and change management are mandatory and expensive for this size plateau
CJ	08	A surprisingly strong influence of the outcome of software projects is the nature of the industry that builds the application
CJ	09	The systems software community has learned the hard way that careful quality control is on the critical path
CJ	10	Military standards are so complex that the productivity of defence application development projects is lower than in any other software sub industry
CJ	11	The MIS (management information systems) community lags in quality control and testing technologies compared to the other communities
CJ	12	As the overall size ranges grow larger, delays and cancellations become more common and also more severe
CJ	13	None of the (six) domains has fully mastered the ability to construct truly large software systems without a significant risk of termination or cancellation
CJ	14	No organisation should tackle software projects above 10.000 FP's without fully evaluating schedules, costs, risks and value
CJ	15	The probability of recovery for a software project in deep distress is fairly low. Prevention is often more effective than control
CJ	16	Successful projects deploy a notable quantity of quality control and project management automation, but tools by themselves do not make successful projects. Capable mangers and capable technical personnel are also needed
Evans et al., 2002		
EV	01	Failure to apply essential project management practices
EV	03	Failure to implement effective software processes
EV	04	Premature victory declarations
EV	06	Untimely decision-making
EV	07	Lack of proactive risk management
K.T. Yeo, 2002		
KY	02	Weak definitions of requirements and scope
KY	03	Inadequate project risk analysis
KY	04	Incorrect assumptions regarding risk analysis
KY	07	Top down management style
KY	08	Poor internal communication
KY	09	Absence of an influential champion and change agent
KY	10	Reactive and not pro-active in dealing with problems

KY	12	Incomplete specifications when project started
KY	13	Inappropriate choice of software
KY	15	Involve high degree of customisation in application
Lorin May, 1998		
LM	02	Stakeholder conflicts
LM	06	Hidden costs of going "lean and mean"
LM	08	Communication breakdowns
LM	10	Late failure warning signals
John S. Reel, 1999		
JR	02	Maintain momentum (attrition, quality and management)
JR	04	Make smart decisions (always use commercial libraries when available)
Jan Oonincx, 1982		
JO	05	Insufficient and/or inefficient use of the possibilities as offered by the computer in the information system
JO	06	Not using the new information system
JO	07	Collection and storage of too much and unsuitable data
JO	09	Set-up of information systems too static
JO	12	Too little attention to the informal aspects in relation to the information system
JO	13	The development process takes too long and the costs are too high
JO	14	Computerisation based on status considerations
JO	17	Insufficient attention to reliability and controllability
Nico Beenker, 2004		
NB	05	Problems escalated too late
NB	06	In cases of failure there is too little appreciation and attention to the quality of the (written) communication
NB	07	Differences in perception
Peter Noordam et al., 2007		
PN	04	Unfamiliarity with scope and complexity (KPMG)
PN	05	Technical complexity and technical integration issues (KPMG)
PN	06	The use of a business case results in a higher degree of satisfaction with the project, whilst the satisfaction with the project is very low when no business case is used
PN	07	The fact whether the costs are evaluated (during and after the project) and whether the benefits are evaluated (at the end of the project) also prove to have a fairly large influence on the success
PN	08	The satisfaction increases when a standard project management methodology is used
PN	10	Technical knowledge is certainly an important skill for project managers to have
PN	13	In larger integrated projects the social complexity is still the largest area of risk
PN	14	Evaluating projects is strongly related to the satisfaction about the successful execution of projects
PN	15	The result favours judging a project not just on whether it has realised the desired functionality on time and within budget, but also favours giving the project responsibility for actually realising the benefits
PN	16	Slavish and blind imitation of project management methods as a guarantee for success often results in a "paper tiger" that negatively influences the progress of the project
Methods, 2007		
JRR	02	The method does not work
TG	01	Don't believe blindly in any one method; use your methods and common sense to measure the reality against your needs

Tarek Abdel-Hamid and Stuart E. Madnick, 1991		
Software Project Dynamics		
Introduction		
TAH	09	System complexity grows as the square of the number of system elements; therefore, experience with a small system cannot account for all the things that will have to be done in a large system. Neither will the Experience Method apply to systems of totally different content {23} *)
TAH	12	The longer an error goes undetected, the more extensive the necessary rework and the greater the cost
TAH	14	The relationship between cost and system size is not linear. In fact, cost increases approximately exponentially as size increases {188}
TAH	16	A major challenge to managers is to motivate employees. A mechanism for motivation, which is attracting interest in the software engineering field, is "goal setting" {49}
Human Resource Management		
TAH	18	Newly hired project members pass through an orientation during which they are less than fully productive {75,243}
Software production / development		
TAH	25	When a project is perceived to be behind schedule, people tend to work harder to bring it back on schedule. There is a threshold beyond which employees would not be willing to work at an "above-normal" rate {169}
TAH	30	Schedule pressures often result in the "overlapping of activities that would have been accomplished better sequentially", and overlapping can significantly increase the chance of errors {252}
System testing		
TAH	31	We assume that undetected errors will become either "Active errors" (errors that produces more errors) or "Passive errors". Because design specs are the blueprints of the system's code, any errors in design will get translated into coding errors
Model Behaviour		
TAH	42	Using the system dynamics model as an experimentation vehicle can provide many benefits to the software engineering community
TAH	43	In addition to permitting less costly and less time consuming experimentation, simulation models make perfectly controlled experiments possible
TAH	44	By using a model of a complex system, more can be learned about internal interactions than would ever be possible through manipulation of the real system
On the accuracy of software estimation		
TAH	52	By imposing different estimates on a software project we create different projects. Research clearly indicates that the decisions that people make in projects and the actions they take are significantly influenced by the pressures and perceptions produced by the project's schedule {57,109,125,224,236}
TAH	54	When the project is perceived to be ahead of schedule, "Parkinson's Law indicates that people will use the extra time for ... personal activities, catching up on the mail, etc. " {49}. This, of course, means that they become less productive
TAH	57	The "Safety Factor Policy" does achieve its intended objective: more accurate estimates. However, the organisation pays dearly for this
Portability of estimation models		
TAH	58	Portability of the models has proven to be especially poor {43,49,144,184}. In the case of a hypothetical software project of 36,000 machine-language executable instructions the highest (of 12) estimate is over 650% higher than the lowest
TAH	61	Different policies affect what the project's cost will end up being and should therefore be explicitly considered when project cost estimates are made
TAH	64	Our model results indicate "adding manpower to a late software project makes it more costly, but *not necessarily later*"

TAH	65	A different distribution of estimated effort among a project's phases creates a different project
Analogy method of software estimation		
TAH	67	Because of the inherent tendency to overshoot, the use of the analogy method in estimating injects a bias in scheduling, a bias that in the long-run generates longer than necessary schedules. The phenomenon of projects consuming longer and longer schedules is one that has been frequently encountered in system dynamics studies of organisational behaviour {241}
The 90% syndrome		
TAH	70	Estimates of the fraction of work completed increase as originally planned until a level of about 80-90% is reached. The programmer's individual estimates then increase only very slowly until the task is actually completed" {28}
TAH	72	The better the measurement tool the earlier it will detect that progress is not keeping up with the underestimated schedule
The economics of quality essurance		
TAH	76	QA policy does have a significant impact on total project cost
Some conclusions		
TAH	77	An integrated approach helps us achieve an overall understanding
TAH	78	The model identifies feedback mechanisms and uses them to structure and clarify relationships in software project management
TAH	79	The schedule overshoot problem can arise not only because of schedule underestimation, but also because of management's hiring policies
TAH	80	A different schedule creates a different project
TAH	81	An important implication is that a software estimation model cannot be judged solely based on how accurately it estimates historical projects
TAH	82	Evidence in the literature indicates that currently available quantitative software estimation tools are not particularly portable from the company in which they were developed to another (e.g., see {43,49}
***Aart van Dijk, 2009** (see sections 9.2 and 10.5)*		
AvD	*01*	*A project leader who works with small teams and who is able to act as working foreman*
AvD	*02*	*A project leader who also looks for projects that appeal to him, also from an application point of view. Preferably not too sizeable but challenging/difficult*
AvD	*03*	*A project leader who rarely has problems with his project officers and manages to motivate them well*
AvD	*04*	*Complete the project properly, it will make you feel good*
AvD	*05*	*Avoid wherever possible "big bang" scenarios*
AvD	*06*	*Make sure that the project officers are enjoying themselves (challenge)*
AvD	*07*	*Involve ICT Management in the project at an early stage (requirements from ICT Management)*
AvD	*08*	*Budget for having a few audits done*
AvD	*09*	*As a project leader, ensure a good project administration but spend the bulk of your time by far on project officers and project aspects with regard to content*
AvD	*10*	*Go for quality and do things right in one single go*

*) {...} = reference number in the list of references included in the book [Abdel-Hamid & Madnick 1991]

7.3 SUCCESS AND FAILURE FACTORS IN ICT PROJECTS: SUFFI TOTAL CHART

Version: 2.0
Date: 1 May 2009
Author: Aart van Dijk

SUccess and Failure Factors in ICT projects (SUFFI Total Chart)

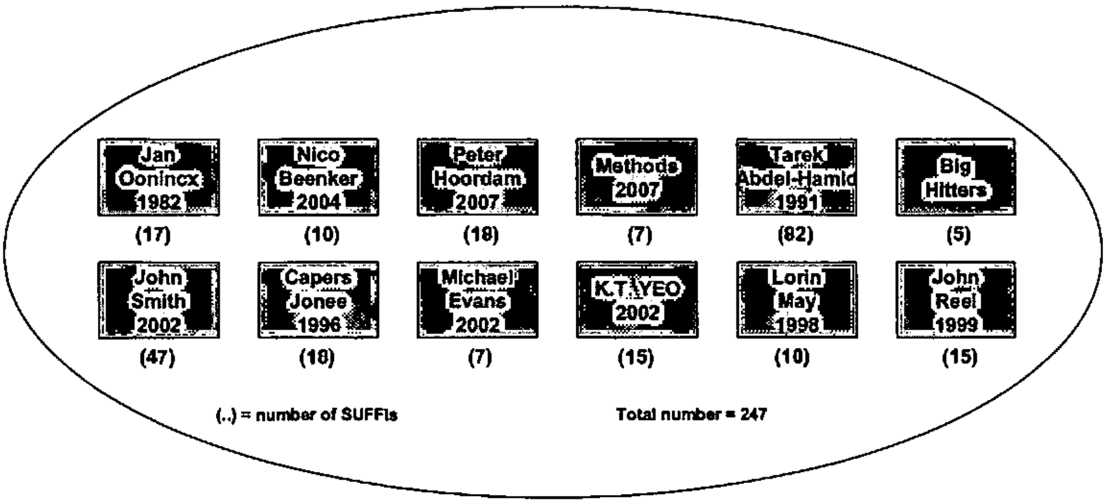


Figure 7.3.1: SUFFI Total Chart

		Version 2.0 - 1 May 2009
SUccess and Failure Factors in ICT projects: SUFFI Total Chart		
Author	**No**	**Description**
***Big Hitters*: Success/failure factors that are mentioned the "Big Hitters" (BH)**		
BH	01	Poor project management
BH	02	Deadlines are unrealistic
BH	03	Poor communication
BH	04	Incomplete/weak definition requirements
BH	05	Insufficient involvement of future users
BH	*06*	*Lack of senior management involvement and commitment (see section 9.8)*
BH	*07*	*Lack of professionalism (see section 9.8)*

John Smith, 2002		
Project conception		
JS	RC01	Project based on an unsound premise or an unrealistic business case
JS	RC02	Buyer failure to define clear project objectives, anticipated benefits and success criteria
JS	RC03	Project based on state-of-the-art and immature technology
JS	RC04	Lack of buyer board-level ownership/commitment or competence
JS	RC05	Buyer's funding and/or time-scale expectations unrealistically low
JS	RC06	Buyer failure to break a complex project into phases or smaller projects
Project initiation/mobilisation		
JS	RC07	Vendor setting unrealistic expectations on cost, time-scale or vendor capability
JS	RC08	Buyer failure to define and document requirements (functional and non-functional)
JS	RC09	Failure to achieve an open, robust and equitable buyer-vendor relationship
JS	RC10	Vendor failure to invest enough resources to scope the project to contract
JS	RC11	Buyer lack of sufficient involvement of eventual end-users
JS	RC12	Vendor underestimation of resources (predominantly person-effort) required
JS	RC13	Vendor failure to define project tasks, deliverables and acceptance processes
JS	RC14	Failure to actively manage risks and maintain robust contingency plans
JS	RC15	Poor project planning, management and execution
JS	RC16	Failure to clearly define roles and responsibilities in the contract/subcontracts
JS	RC17	Full-scope, fixed-price contracting (requirements, design and development)
System design		
JS	RC18	Failure to 'freeze' the requirements baseline and apply change control
JS	RC19	Poor choice of technical platform and/or architecture
JS	RC20	Vendor starting a phase prior to completing a previous phase
JS	RC21	Poor choice of design/development method
JS	RC22	Failure to undertake effective project reviews and take decisive action
JS	RC23	Vendor lack/loss of skilled resources
JS	RC24	Poor vendor standards deployment (design, coding, testing, configuration management, etc.)
JS	RC25	Poor vendor requirements traceability (requirements > design > code > test)
JS	RC26	Buyer retention of design authority with right to approve/reject low-level designs
System development		
JS	RC27	Delays causes the project to be overtaken by advances in technology
JS	RC28	Vendor failure to 'freeze' the design (and technical platform) and apply change control
JS	RC29	Inadequate vendor training and supervision of junior staff
JS	RC30	Inadequate vendor review of designs/code/documentation
JS	RC31	Poor vendor management of sub-contractors
JS	RC32	Lack of a formal, 'engineering' approach to integration and testing by vendor
JS	RC33	Insufficient attention paid by vendor to non-functional requirements
System implementation		
JS	RC34	Buyer failure to manage the change implicit in the project (people, processes, technology)
JS	RC35	Inadequate user/systems training
JS	RC36	Catastrophic failure of the system, with no effective contingency arrangement
JS	RC37	Missing a crucial 'go live' date
System operation, benefit delivery, stewardship and disposal		
JS	RC38	Buyer failure to measure actual delivered benefit and take corrective action
JS	RC39	Buyer failure to maintain/enhance system post-implementation
JS	RC40	Changes in the competitive or macro-economic environment
***John Smith, 2002* (UK public sector)**		
JS	PUBRC01	Lack of senior management involvement and commitment
JS	PUBRC02	Failure to focus on key business and end-user needs
JS	PUBRC03	Failure to break complex projects into manageable, separately contracted 'chunks'

JS	PUBRC04	Poor and unimaginative project management
JS	PUBRC05	Poor risk management and contingency planning
JS	PUBRC06	Unclear contracts and poor contract management
JS	PUBRC07	Insufficient focus on user training needs and the design of training interventions
Capers Jones, 1996		
		Large software systems (FP = Function Point)
CJ	01	1.000 FP's: Quality control is a major requirement at this size range
CJ	02	1.000 FP's: With team development (up to 10 staff members), issues of system segmentation and interfaces among components become problematic
CJ	03	10.000 FP's: This size range is often plagued by cost and schedule overruns and by outright cancellations
CJ	04	10.000 FP's: Development teams of 100 or so are common, so communication and interface problems are endemic
CJ	05	10.000 FP's: Configuration control and change management are mandatory for this size plateau
CJ	06	100.000 FP's: Development teams number in the hundreds, often in multiple locations that may even be in different countries. Communication problems are rampant
CJ	07	100.000 FP's: Formal configuration control and change management are mandatory and expensive for this size plateau
CJ	08	A surprisingly strong influence of the outcome of software projects is the nature of the industry that builds the application
CJ	09	The systems software community has learned the hard way that careful quality control is on the critical path
CJ	10	Military standards are so complex that the productivity of defence application development projects is lower than in any other software sub industry
CJ	11	The MIS (management information systems) community lags in quality control and testing technologies compared to the other communities
CJ	12	As the overall size ranges grow larger, delays and cancellations become more common and also more severe
CJ	13	None of the (six) domains has fully mastered the ability to construct truly large software systems without a significant risk of termination or cancellation
CJ	14	No organisation should tackle software projects above 10.000 FP's without fully evaluating schedules, costs, risks and value
CJ	15	The probability of recovery for a software project in deep distress is fairly low. Prevention is often more effective than control
CJ	16	Successful projects deploy a notable quantity of quality control and project management automation, but tools by themselves do not make successful projects. Capable mangers and capable technical personnel are also needed
Evans et al., 2002		
Seven characteristics of dysfunctional software projects		
EV	01	Failure to apply essential project management practices
EV	02	Unwarranted optimism and unrealistic management expectations
EV	03	Failure to implement effective software processes
EV	04	Premature victory declarations
EV	05	Lack of program management leadership
EV	06	Untimely decision-making
EV	07	Lack of proactive risk management
K.T. Yeo, 2002		
Critical failure factors in information system projects		
Process driven issues (business planning, project planning, project management and control)		
KY	01	Underestimate of timeline
KY	02	Weak definitions of requirements and scope

KY	03	Inadequate project risk analysis
KY	04	Incorrect assumptions regarding risk analysis
KY	05	Ambiguous business needs and unclear vision
Context driven issues(corporate culture, corporate management, users, politics)		
KY	06	Lack user involvement and inputs from the onset
KY	07	Top down management style
KY	08	Poor internal communication
KY	09	Absence of an influential champion and change agent
KY	10	Reactive and not pro-active in dealing with problems
Content driven issues (information technology, business process and system design, IT/IS professional and knowledge sources)		
KY	11	Consultant/vendor underestimated the project scope and complexity
KY	12	Incomplete specifications when project started
KY	13	Inappropriate choice of software
KY	14	Changes in design specifications late the project
KY	15	Involve high degree of customisation in application
Lorin May, 1998		
The failure causes are listed in no particular order		
LM	01	Poor user input
LM	02	Stakeholder conflicts
LM	03	Vague requirements
LM	04	Poor cost and schedule estimation
LM	05	Skills that do not match the job
LM	06	Hidden costs of going "lean and mean"
LM	07	Failure to plan
LM	08	Communication breakdowns
LM	09	Poor architecture
LM	10	Late failure warning signals
John S. Reel, 1999		
The five essential/critical factors to managing a successful software project		
JR	01	Start on the right foot (nr 06 -15)
JR	02	Maintain momentum (attrition, quality and management)
JR	03	Track progress (a large problem in managing software development is figuring where you are in your schedule)
JR	04	Make smart decisions (always use commercial libraries when available)
JR	05	Institutionalise post-mortem analyses (if you do not take time to figure out what happened during a project, both the good and the bad, you are doomed to repeat it)
Tom Field [55] gave 10 signs of IS project failure		
JR	06	Project managers don't understand users' needs
JR	07	The project's scope is ill-defined
JR	08	Project changes are managed poorly
JR	09	The chosen technology changes
JR	10	Business needs change
JR	11	Deadlines are unrealistic
JR	12	Users are resistant
JR	13	Sponsorship is lost
JR	14	The project lacks people with appropriate skills
JR	15	Managers ignore best practices and lessons learned
Jan Oonincx, 1982		
JO	01	The lack of a methodology for developing information systems
JO	02	Insufficient insight into decision-making processes and information requirements

JO	03	The feasibility of integral information systems
JO	04	Insufficient involvement and cooperation of users in setting up information systems
JO	05	Insufficient and/or inefficient use of the possibilities as offered by the computer in the information system
JO	06	Not using the new information system
JO	07	Collection and storage of too much and unsuitable data
JO	08	Ineffectiveness and limited applications of traditional systems for data storage
JO	09	Set-up of information systems too static
JO	10	Insufficient suitable human resources for development and implementation of information systems
JO	11	The influence of changes to the information system on the organisation
JO	12	Too little attention to the informal aspects in relation to the information system
JO	13	The development process takes too long and the costs are too high
JO	14	Computerisation based on status considerations
JO	15	Too little attention to the aspect of profitability and the lack of measuring instruments for determining the benefits of the information system
JO	16	Unsuitable equipment and software
JO	17	Insufficient attention to reliability and controllability
Nico Beenker, 2004		
NB	01	Planning ahead too optimistic
NB	02	Badly phrased contracts
NB	03	Poor project management
NB	04	Poor communication
NB	05	Problems escalated too late
Next five aspects are perhaps the basis for a tool using which success and failure can be more strongly influenced as compared to the usual professional tools		
NB	06	In cases of failure there is too little appreciation and attention to the quality of the (written) communication
NB	07	Differences in perception
NB	08	One or more existing misfits between supplier and client
NB	09	Commercial dynamics (the collection of all undesired effects involved in commercial outsourcing of a valuable task)
NB	10	Type of partnership. People (supplier and client) often think differently about cooperation
Peter Noordam et al., 2007		
PN	01	Poor project management (KPMG)
PN	02	Lack of communication in and around the project (KPMG)
PN	03	Objectives not defined (KPMG)
PN	04	Unfamiliarity with scope and complexity (KPMG)
PN	05	Technical complexity and technical integration issues (KPMG)
PN	06	The use of a business case results in a higher degree of satisfaction with the project, whilst the satisfaction with the project is very low when no business case is used
PN	07	The fact whether the costs are evaluated (during and after the project) and whether the benefits are evaluated (at the end of the project) also prove to have a fairly large influence on the success
PN	08	The satisfaction increases when a standard project management methodology is used
PN	09	It is sensible to set up projects such that these do not exceed the 1.5 million dollar line in order to increase the chance of success (Standish Group International Inc.)
PN	10	Technical knowledge is certainly an important skill for project managers to have
PN	11	Cyclic execution of risk management results in the largest amount of satisfaction
PN	12	Technical complexity is still underestimated in IT projects
PN	13	In larger integrated projects the social complexity is still the largest area of risk

PN	14	Evaluating projects is strongly related to the satisfaction about the successful execution of projects
PN	15	The result favours judging a project not just on whether it has realised the desired functionality on time and within budget, but also favours giving the project responsibility for actually realising the benefits
PN	16	Slavish and blind imitation of project management methods as a guarantee for success often results in a "paper tiger" that negatively influences the progress of the project
Methods, 2007		
JRR	01	The method does not work
JRR	02	The designer designs and not the method
TG	01	Don't believe blindly in any one method; use your methods and common sense to measure the reality against your needs
KJM	01	Success or failure of a software project is all about management
KJM	02	Selection of a particular software development method has major consequences for the management of a project
KJM	03	Selection of a different method is not always the solution but different application of a method is
KJM	04	Project management is risk management
Tarek Abdel-Hamid and Stuart E. Madnick, 1991		
Software Project Dynamics		
Introduction		
TAH	01	Software engineering encompasses both the technical aspects of software development as well as the managerial ones {33,47,246} *)
TAH	02	Poor management can increase software costs more rapidly than any other factor {261}
TAH	03	We still lack the fundamental understanding of the software development process {87,96,173} and without such an understanding the likelihood of any significant gains in the management of software development front is questionable {36,130,161}
TAH	04	There are hundreds of variables that affect software development. Furthermore, these variables are not independent; many of them are related to one another {119,189}. THA constructed *a holistic model* of the software development process
TAH	05	The behaviour of an individual subsystem in isolation may be different from its behaviour when it interacts with other subsystems {68}
TAH	06	Brooks' Law: adding more people to a late software project makes it later {57}
TAH	07	People under time pressure don't work better, they just work faster. ... In the struggle to deliver any software at all, the first casualty has been consideration of the quality of the software delivered {81}
TAH	08	Two factors significantly influence the initial estimate of the job size: (1) the firm's previous experience; and (2) the general over-all tendency to underestimate the job size {49,223}
TAH	09	System complexity grows as the square of the number of system elements; therefore, experience with a small system cannot account for all the things that will have to be done in a large system. Neither will the Experience Method apply to systems of totally different content {23}
TAH	10	In a multi-project environment, competition for company resources becomes a significant dimension
TAH	11	It is difficult to measure performance in programming {181}
TAH	12	The longer an error goes undetected, the more extensive the necessary rework and the greater the cost
TAH	13	Inadequate planning is the primary reason for loss of control on many computer programming projects {208}
TAH	14	The relationship between cost and system size is not linear. In fact, cost increases approximately exponentially as size increases {188}

TAH	15	Frequent budget manipulations by management to avoid overruns make historical cost data questionable {188}
TAH	16	A major challenge to managers is to motivate employees. A mechanism for motivation, which is attracting interest in the software engineering field, is "goal setting" {49}
TAH	17	All programmers are optimists. They always assume that "This time it will surely run" or "I just found the last bug" {57}
Human Resource Management		
TAH	18	Newly hired project members pass through an orientation during which they are less than fully productive {75,243}
TAH	19	The training of newcomers, both technical and social, is usually carried out by the "old-timers". This is costly because "while the old-timer is helping the new employee learn the job, his own productivity on his other work is reduced {55,74,77}
TAH	20	Toward the end of the project there is likely to be reluctance to bring in new people
Software production / development		
TAH	21	Under schedule pressures, walk throughs and inspections are usually the greatest casualties, they are not only relaxed but often suspended altogether {94}
TAH	22	Problems of communication and motivation are responsible for inadequacies in process and for consequent losses in productivity {240}
TAH	23	Work force experience level and increases in project familiarity due to learning-curve effects affect the productivity of a software development project {76,234,261}
TAH	24	The project goals and schedules can play a significant motivational role throughout the life of a software development project
TAH	25	When a project is perceived to be behind schedule, people tend to work harder to bring it back on schedule. There is a threshold beyond which employees would not be willing to work at an "above-normal" rate {169}
TAH	26	When project members perceive some "excesses" in the schedule (the case of negative schedule pressure), some excesses will be "absorbed" by the workers as "under-work" before downward adjustments are made in the project's schedule {49,131}. As with positive schedule pressures, there are limits on how "fat" employees are willing or allowed to absorb
TAH	27	Communication overhead increases in proportion to n^2, where n is the size of the team {57,180,231,236,273}
TAH	28	Design errors in the early design phase are generated at a higher rate than are coding errors {165}
TAH	29	Newly hired employees are not only less productive on average but also more error-prone than their experienced counterparts {92,189}
TAH	30	Schedule pressures often result in the "overlapping of activities that would have been accomplished better sequentially", and overlapping can significantly increase the chance of errors {252}
System testing		
TAH	31	We assume that undetected errors will become either "Active errors" (errors that produces more errors) or "Passive errors". Because design specs are the blueprints of the system's code, any errors in design will get translated into coding errors
TAH	32	The earlier the undetected error is, the more "generations" of errors it will produce, and thus the more costly it will end up being
Controlling / planning		
TAH	33	Progress in a software project is measured by the number of resources consumed, tasks completed, or both
TAH	34	Progress, especially in the earlier phases of software development, is measured by the rate of expenditure of resources rather than by some count of accomplishments {81}
TAH	35	As a software project develops, project members often realise that they have under-estimated the number of tasks that constitutes the software system being developed {61}
TAH	36	There is a powerful tendency to focus on the highly visible mainline components of the software, and to underestimate or completely miss the unobtrusive components (e.g., help message processing, error processing, and moving data around {49}

TAH	37	In some cases the management becomes increasingly willing to "pay any price" necessary to avoid overshooting the "Maximum Tolerable Completion Date". In such cases, management is often willing to hire more people
Case		
TAH	38	At the System Development Section 25% of an experienced employee's time is committed per new employee
TAH	39	Actual productivity rarely equals potential productivity because of losses from communication and motivation problems
TAH	40	The model accurately portrays management's inclination not to adjust the project's scheduled completion date during most of the development phase. Adjustments are instead made in the project's work force level. This behaviour pattern arises, according to DeMarco, for political reasons
TAH	41	As the date approached, pressures developed that overrode the work force stability: management became increasingly willing to "pay any price" necessary to avoid overshooting the "Maximum Tolerable Completion Date"
Model Behaviour		
TAH	42	Using the system dynamics model as an experimentation vehicle can provide many benefits to the software engineering community
TAH	43	In addition to permitting less costly and less time consuming experimentation, simulation models make perfectly controlled experiments possible
TAH	44	By using a model of a complex system, more can be learned about internal interactions than would ever be possible through manipulation of the real system
TAH	45	A full-time employee allocates, on the average, 60% of his or her time to productive work on the project
TAH	46	Most software projects start with a smaller core of designers, and as the project develops, the work force slowly builds up to higher levels
TAH	47	In the early stages of a project, project managers are generally willing to adjust the work force level to maintain the project on its scheduled course. However, as the project proceeds, management becomes increasingly reluctant to add new people out of an increasing desire that the work force stabilise
TAH	48	The shift away from work force adjustments to schedule adjustments continues as the project progresses
TAH	49	Project members are not willing to maintain an above-normal work rate indefinitely. Once people start working at a rate above their normal rate, their "Overwork Duration Threshold" decreases because people enjoy and need their slack time
On the accuracy of software estimation		
TAH	50	For many years, estimation of software project development time and cost had been an intuitive process based on experience and analogy {200}
TAH	51	Having captured within our integrative system dynamics model "influence variables of software development and their causal relationships", we embark on a quantitative analysis of software cost and schedule estimation
TAH	52	By imposing different estimates on a software project we create different projects. Research clearly indicates that the decisions that people make in projects and the actions they take are significantly influenced by the pressures and perceptions produced by the project's schedule {57,109,125,224,236}
TAH	53	A higher work force level generally means more communication and training overhead, which in turn affect productivity negatively. Scheduling can dramatically change the manpower loading throughout the life of a project
TAH	54	When the project is perceived to be ahead of schedule, "Parkinson's Law indicates that people will use the extra time for … personal activities, catching up on the mail, etc." {49}. This, of course, means that they become less productive
TAH	55	The management reserves ranged from 5% to 50% of the estimated software cost with a mean of 18% {85}
TAH	56	The project's initial estimates create pressures and perceptions that affect how people behave on the project. Overestimating a project often leads to an expansion of the project members' slack time activities, which leads to further reductions in productivity

TAH	57	The "Safety Factor Policy" does achieve its intended objective: more accurate estimates. However, the organisation pays dearly for this
Portability of estimation models		
TAH	58	Portability of the models has proven to be especially poor {43,49,144,184}. In the case of a hypothetical software project of 36,000 machine-language executable instructions the highest (of 12) estimate is over 650% higher than the lowest
TAH	59	A software cost estimation model should avoid the use of variables whose values cannot be determined until the project is complete
TAH	60	The policy of allocating project members half-time to the project results in a cost that is about 22% higher
TAH	61	Different policies affect what the project's cost will end up being and should therefore be explicitly considered when project cost estimates are made
TAH	62	More people on the project means more work gets done. It also means that the project team's overall productivity is lower because of the increased communication and training overheads
TAH	63	Variables used in cost estimation tend to be those which are easier to measure, quantify, and estimate, even if they are not the most significant {65}
TAH	64	Our model results indicate "adding manpower to a late software project makes it more costly, but *not necessarily later*"
TAH	65	A different distribution of estimated effort among a project's phases creates a different project
Analogy method of software estimation		
TAH	66	A project's estimate creates pressures and perceptions that directly influence the decisions people make and the actions they take throughout the project's life cycle
TAH	67	Because of the inherent tendency to overshoot, the use of the analogy method in estimating injects a bias in scheduling, a bias that in the long-run generates longer than necessary schedules. The phenomenon of projects consuming longer and longer schedules is one that has been frequently encountered in system dynamics studies of organisational behaviour {241}
TAH	68	When a software development project overruns its schedule there is an apparent cause: the project was poorly estimated
TAH	69	After all, software estimation is not yet an exact science. Significantly, it is often impossible in a real life situation to demonstrate that underestimation was *not* in fact the cause
The 90% syndrome		
TAH	70	Estimates of the fraction of work completed increase as originally planned until a level of about 80-90% is reached. The programmer's individual estimates then increase only very slowly until the task is actually completed" {28}
TAH	71	The "90% syndrome" arises because of the interaction of two factors: underestimation and imprecise measurement of progress
TAH	72	The better the measurement tool the earlier it will detect that progress is not keeping up with the underestimated schedule
TAH	73	The result of sticking with a schedule that is too tight is often an increase in the project's cost {49} due to a large work force level
The economics of quality assurance		
TAH	74	A significant feature of the relationship between the QA effort expended and the percentage of errors detected during development, is the diminishing returns as QA expenditures exceed 20-30% of development effort
TAH	75	QA is used not only in the development phase but also to minimise the cost of the testing phase
TAH	76	QA policy does have a significant impact on total project cost
Some conclusions		
TAH	77	An integrated approach helps us achieve an overall understanding
TAH	78	The model identifies feedback mechanisms and uses them to structure and clarify relationships in software project management
TAH	79	The schedule overshoot problem can arise not only because of schedule underestimation, but also because of management's hiring policies

TAH	80	A different schedule creates a different project
TAH	81	An important implication is that a software estimation model cannot be judged solely based on how accurately it estimates historical projects
TAH	82	Evidence in the literature indicates that currently available quantitative software estimation tools are not particularly portable from the company in which they were developed to another (e.g., see {43,49})
***Aart van Dijk, 2009** (see sections 9.2 and 10.5)*		
AvD	*01*	*A project leader who works with small teams and who is able to act as working foreman*
AvD	*02*	*A project leader who also looks for projects that appeal to him, also from an application point of view. Preferably not too sizeable but challenging/difficult*
AvD	*03*	*A project leader who rarely has problems with his project officers and manages to motivate them well*
AvD	*04*	*Complete the project properly, it will make you feel good*
AvD	*05*	*Avoid wherever possible "big bang" scenarios*
AvD	*06*	*Make sure that the project officers are enjoying themselves (challenge)*
AvD	*07*	*Involve ICT Management in the project at an early stage (requirements from ICT Management)*
AvD	*08*	*Budget for having a few audits done*
AvD	*09*	*As a project leader, ensure a good project administration but spend the bulk of your time by far on project officers and project aspects with regard to content*
AvD	*10*	*Go for quality and do things right in one single go*

*) {...} = reference number in the list of references included in the book [Abdel-Hamid & Madnick 1991]

7.4 DEFINITION OF THE PROBLEM [Answer to sub question 6]

Definition of the problem, sub question 6:

Is it possible to present the SUFFIs in a SUFFI Chart that is easy to use in practice by others?

Answer:

In the previous chapters, the **SUFFI model** was constructed. It consists of tables 3.7.11, 3.7.12, 3.8.2 and 4.2. The SUFFI model can be put to use immediately and next further tested and improved. In order to simplify its use even more, the tables have been combined into one single table, the so-called **SUFFI Chart**. This is included in section 7.2.

The SUFFI Chart was created from a number of tables (3.7.11, 3.7.12, 3.8.2 and 4.2). These tables include the research results of various researchers. In order to be able to test new research results more easily against all tables, a **SUFFI Total Chart** was created in which all tables (3.7.1-3.7.10, 3.8.1 and 4.2) were included.

CHAPTER 8

CASE "NETHERLANDS COURT OF AUDIT"

8.1 INTRODUCTION

It has been indicated that the SUFFI model or the SUFFI-Chart as shown in chapter 7 can be used immediately and that further investigation may take place. For reasons of major social interest, the subject of success/failure factors will demand necessary attention over the next few years (see chapter 9). Both researchers as well as project managers are invited to test the SUFFI-Chart against their own research results and practical experience. This will enable further improvement of the SUFFI-Chart, serving all those involved in ICT projects well.

At the end of November 2007, the Netherlands Court of Audit published the report "Lessen uit ICT-projecten bij de overheid – Deel A" [Wijsman et al. 2007] (Lessons from government ICT projects – Part A). Various experts have commented on this report. It is interesting to test the SUFFI-Chart against this case. Section 8.2 provides a brief summary of the report. Section 8.3 includes the following tables:

- table 8.1: main success/failure factors regarding ICT projects according to the report of the Court of Audit;
- table 8.2: recommendations/remarks from the experts as a result of the report of the Court of Audit;
- table 8.3: overview Advisors/Experts.

Table 8.1 includes the conclusions of the Court of Audit represented by 39 success/failure factors. In table 8.2, the recommendations/remarks of experts have been recorded in the shape of 58 items of recommendations/remarks. For all 97 items, it was checked which SUFFIs from the SUFFI-Chart can be related to the item in question. For each item, one or more related SUFFIs were included. In this, the main applicable SUFFIs have been shown. Completeness was not the aim. In section 8.4, the conclusions of this testing of the SUFFI-Chart against the case come up for discussion.

8.2 REPORT COURT OF AUDIT

Conclusion

As far as ICT projects are concerned, the government handles these badly. The Court of Audit arrived at this conclusion in the research report that appeared at the end of November 2007 [Wijsman et al. 2007].

What?

The research report of the Court of Audit confirmed that the Dutch government spends billions every single year on ICT projects that fail entirely or in part. The research was executed as a result of questions asked in the Dutch Lower Chamber earlier this year. It is not clear how many projects and how much money are/is involved. Minister Guusje ter Horst (Ministry of the Interior) endorsed the findings as far as the basics are concerned.

What went wrong?

The list of (partly) failed government ICT projects is a long one. The report of the Court of Audit mentioned, for example, a project for the Department of Immigration and Naturalisation, in which too many different organisations were involved and the outsourcing of the P-Direkt project, which according to the Court of Audit failed because of a succession of shortcomings in decision-making, steering and control on the side of the government. The report also gave attention to the C2000 project and the integrated incident room system of the police, which are said to have suffered from poor project management and information provision.

How did it happen?

The research by the Court of Audit showed that government ICT projects often become far too ambitious and complex through a combination of political, organisational and technical factors. According to the researchers, these projects lacked in balance between ambition, available manpower, means and time. The reasons behind this being the various different interests of a Minister, the Lower Chamber and the ICT suppliers. All three could benefit from ambitious projects, in order to solve "major problems by means of major solutions". The Court of Audit suspects that this creates a spiral that ends in a complex project with the status of a political fact, which means that there is no elegant way back. Furthermore, the research showed that there was often insufficient information on ICT projects. The Minister needs information in order to be able to control properly; the Lower Chamber needs the information to be able to check the Minister. Often, no information was provided on operationalised policy objectives.

How can it be remedied?

According to the Court of Audit, it is up to the ministries to break out of the spiral in which projects become too complex. After all, the civil servants are the ones that commissioned the projects and they are in regular contact with the ICT suppliers. At the ministry, they have to be able to resist the dynamics in which a project becomes increasingly more ambitious. The Court of Audit stresses that sufficient realism and being more in control are of major importance. According to the researchers, realism includes being aware of the fact that ICT is not a quick fix for a problem, that political deadlines can be fatal to a project and that ICT ambitions also include a gap between policy and execution. Moreover, it should be possible to reconsider decisions prematurely and muddling on might be avoided through an exit strategy.

According to the Court of Audit, ministries will get more grip on projects when the Minister is viewed as a fully-fledged conversation partner for the Lower Chamber and the ICT supplier, when decisions are made phased and on the basis of well-founded plans and when it is possible to reconsider.

Could it not have been done differently?

The Court of Audit pointed out that this has been done before in the United States. Over there, 1996 saw the drawing up of the Clinger Cohen Act, a federal law that records that the American government has to observe certain principles when outsourcing ICT projects. All budget holders in the American government are obliged to use a portfolio approach when making ICT investments, the same as asset managers and fund managers in the financial markets do.

In line with Clinger Cohen, the Court of Audit advised to submit the relevance of every new system to "a test against the core processes of the body in question, look at the coherence with other systems and to critically investigate better and existing alternatives". A cost-benefit analysis also needs to be made every time and an assessment with regard to the total ICT portfolio of the organisation.

Professor Chris Verhoef, Professor at the Free University in Amsterdam, stated in a publication [Verhoef 2007] that because of laws such as the Clinger Cohen Act, the government should no longer assess ICT projects as isolated things. "In that case these simply have to be included in coherence with other affairs and have to be assessed as is done with other decisions; considering financial-economic arguments."

Minister Ter Horst has stated that improvement of the way ICT is dealt with, is worked at, as part of the government's operational management.

8.3 TESTING OF THE CASE AGAINST THE SUFFI CHART

Conclusions Court of Audit

No.	Success/failure factors	Related SUFFIs
AR1	Be realistic in your ambition	BH/02 JS/RC01
AR2	ICT is not a 'quick fix' for a problem. The decision-makers in the political field believe ICT to be a miracle cure for solving all sorts of policy issues	JS/RC01 JS/RC02
AR3	Political deadlines may be fatal to a project	BH/02
AR4	ICT ambitions (also) include a gap between policy and execution	JS/RC01 JS/RC02
AR5	Reconsiderations in mid-stream are often inevitable	JS/RC22 LM/10
AR6	Muddling on can be avoided by means of an exit strategy	JS/RC22
AR7	Make sure to have a grip on your ICT projects	BH/01 JS/RC15
AR8	The Minister should be a fully-fledged conversation partner both for the Lower Chamber and for the ICT supplier	JS/PUBRC01 JS/RC16
AR9	Decision-making should take place phased	BH/03 JS/RC20 JS/RC22
AR10	It is necessary to make decisions on the basis of well-founded plans and that projects are assessed within the entire project portfolio (business case)	JS/RC01 JS/RC02
AR11	Government ICT projects are often too complex	JS/PUBRC03
AR12	Organisational complexity is created through involvement of different autonomous organisations	JS/PUBRC01 JS/PUBRC06 JS/RC16
AR13	Individual organisations often pursue their own objectives instead of shared chain goals	JS/PUBRC01 JS/PUBRC04 JS/RC01 JS/RC02
AR14	When realising political objectives, the relation between ICT and organisations is often overlooked	JS/PUBRC01 JS/RC01
AR15	What often makes ICT projects complex in a different way is the fact that ICT systems are relatively rigid. As opposed to many political and organisational processes, which are dynamic and flexible.	BH/04 JS/RC01 JS/RC02
AR16	Changes 'in midstream' are often difficult to realise from a technical point of view	BH/04 JS/RC18 JS/RC19
AR17	ICT projects become too complex because of the combination of political, organisational and technical factors	BH/04 TAH/09
AR18	In complex projects there is no balance between ambition, available manpower, means and time	BH/01 BH/04 JS/RC15
AR19	Good information is required for good grip on (the execution of) ICT projects. That concerns both the Minister responsible as well as the Lower Chamber	BH/03 NB/06

AR20	Ministers do like to show decisiveness towards the Lower Chamber: decisiveness is made visible through presenting an ambitious project and by linking this with a concrete and short deadline	BH/02
AR21	In order to survive, ICT suppliers need orders, preferably substantial ones	BH/01 JS/RC01 JS/RC02 JS/RC20
AR22	The actors are all 'naturally' inclined to think of big solutions to big problems	JS/RC01 JS/RC02 JS/RC09 JS/RC20
AR23	The actors do not counterbalance each other	JS/PUBRC01 JS/RC09
AR24	A Minister needs policy information for being able to control properly	BH/03 NB/06
AR25	Good information is required for good grip on (the execution of) ICT projects.	JS/RC15 NB/06
AR26	The position and responsibilities of a Minister differ per project	BH/01 JS/PUBRC01 JS/RC16
AR27	The report on an ICT investment should always include a number of obligatory subjects:	JS/RC01 JS/RC02
AR28	The relevance of the new system to the core processes	JS/RC01 JS/RC02
AR29	The coherence with other systems	JS/RC01 JS/RC02
AR30	The question whether there is no better – possibly existing – alternative	JS/RC01 JS/RC02
AR31	Reflection on the business processes in order to improve these if possible	JS/PUBRC02 JS/RC02
AR32	A cost-benefit analysis	JS/RC01 JS/RC07 TAH/14 TAH/43
AR33	An assessment of the project with regard to the entire IT portfolio of the body in question	JS/RC01 JS/RC02 TAH/77
AR34	There is no central administration of ICT projects or of their failure	JS/PUBRC01 NB/06
AR35	The ICT used by the police requires improvement in various parts of the programme control	JS/RC01 TAH/77
AR36	In the case of P-Direkt, the outsourcing of the ICT component failed partly as a consequence of insufficient control	JS/PUBRC01 JS/PUBRC04
AR37	ICT systems often need to be connected to other, often existing, systems	JS/RC01 JS/RC02 TAH/77
AR38	A project can be made manageable by dividing it into smaller sub projects that are easier to manage	JS/PUBRC03 JS/RC06
AR39	Each of the actors, Ministers, Lower Chamber and ICT suppliers benefits, as such for legitimate reasons, for wanting large and ambitious projects	JS/RC01 JS/RC02 TAH/77

Table 8.1: Main success/failure factors in ICT projects according to the report produced by the Court of Audit

Remarks/Recommendations from experts

No.	Remarks/Recommendations	Advisor	Related SUFFIs
	Remarks		
	Important points as overlooked by the Court of Audit according to the experts:		
EX1	*Political inability*: Chamber and government have no idea of the impact and practicability of the rules they come up with	JS	JS/PUBRC01 JS/RC01 JS/RC05 JS/RC06
EX2	*Political inability*: Unclear processes, vague agreements and compromises cannot be computerised	JW	BH/04 JS/RC02 JS/RC04
EX3	*Lack of transparency*: Failing projects are covered with the cloak of political charity	JS	BH/03 JS/PUBRC04 NB/06
EX4	*Insufficient attention to architecture*: Architecture is crucial for getting IT under control and for carrying out commercial customership	DR	JS/RC19
EX5	*Failing project management*: Extension is never to be blamed on inexperienced employees, new technologies, shifting specifications but is always and exclusively caused by weak management	DR	BH/01 JS/PUBRC04 PN/10
EX6	*Failing project management:* Through a lack of knowledge and experience and insufficient problem-solving capacity, small problems develop into major problems	HD	BH/01 JS/RC23 PN/10
EX7	*Failing IT expertise*: Through a lack in technical knowledge as regards to content, there is too much dependency on externals that often have their own agenda surrounding the projects	HD JS	PN/05 PN/10
EX8	*Opportunism in IT companies*: IT suppliers are too eager to build solutions that would really be advised against because the customer is not yet ready	DR	JS/RC07 JS/RC12 PN/05
EX9	*Opportunism in IT companies:* Lesson one is to reorganise first and only then to computerise. AvD: This was as early as the start of the seventies taught by Professor Brussaard at the TU Delft!	DR HN	JS/PUBRC02 JS/RC02
EX10	*Unclear allocatian of 'project property'*: Negative competency struggle, often of the type 'that is not my responsibility'	HN	JS/PUBRC01 JS/RC04 JS/RC16
EX11	*Uncleor allocation of 'project property'*: Too many people are interfering in the projects, which is why one gets bogged down in side issues	GS	JS/RC02 JS/RC16
EX12	*Unclear allocation of 'project property'*: (hidden) political unwillingness for cooperation	JS	JS/RC01 JS/RC02
EX13	*Increasing juridification of outsourcing*: For that reason, client and supplier have insufficient elbow room within the execution for diverging from the original task, even though this would objectively be better for the client	HD	JS/RC08 JS/RC17

	Recommendations		
EX21	Before computerising, optimise and simplify your work processes first	JW	JS/PUBRC02 JS/RC02
EX22	Work under IT architecture: ensure coherence, both in the functional as well as the technical field	JS	JS/RC19 TAH/77
EX23	A good IT architect will start at established 'best practices' and proven technology	DR	JS/RC19
EX24	With regard to ICT, the government will have to make choices, in short will have to apply portfolio management	JS, CV	JS/PUBRC01 JS/RC01 JS/RC02 TAH/77
EX25	Portfolio management is best based on the business economics rule (Net Present Value and Return On Investment)	CV	JS/RC01 JS/RC02 JS/RC06 TAH/14
EX26	Information technology is expensive, very expensive because it is all handmade	CV	JS/RC01 JS/RC02 JS/RC06 TAH/14
EX27	A methodically underpinned and economically translated estimation of the project risk is necessary	CV	EV/07 JS/PUBRC05 JS/RC14
EX28	Dare to stop failures in time	DR	BH/01 NB/05 NB/06 JS/RC22
EX29	The larger the project, the more chance of failure. All risks that are usually manageable will become failure factors (as the projects become larger)	DR	JS/PUBRC03 JS/RC02 TAH/09
EX30	Projects should become less substantial	Var	JS/PUBRC03 JS/RC02 TAH/09
EX31	In the case that the ambitions of politicians cannot be restricted, the commissioned projects will have to be divided into bite size modules. The principles of architecture may offer guidance when modulising	CV	JS/PUBRC03 JS/RC02 JS/RC19 TAH/09
EX32	The (state)government should prevent having to do things twice	DR	JS/RC19 TAH/77
EX33	Monitor the scope of projects, avoid 'scope creep'	Var	JS/RC18
EX34	Implement a government-wide method for project management	Var	BH/01 PN/08
EX35	Ensure a certified system for quality management	Var	TAH/74 TAH/75 TAH/76
EX36	Raise the possibility of consultation between suppliers in advance. Precompetitive consultation should be the norm instead of the exception	Var HD	JS/RC09 JS/RC31
EX37	For the most part, failure factors are usually and certainly to be found in the definition of a problem and the deadlines and budgets	HN	BH/01 JS/RC01
EX38	The government is obliged to have a countercheck carried out but this does not happen	CV	JS/RC22 JS/RC30

EX39	It is of major importance to have independent experts performs audits on a regular basis, during the execution as well. This is a task for the Court of Audit	Var JS	PN/14 JS/RC22
EX40	Place project risks at the suppliers. Suppliers may be expected to have an enterprising mentality	Var HD	JS/RC09
EX41	Problems concerning large computerisation projects always boil down to the same thing: insufficient communication	RHG	BH/03
EX42	The applicant often does not know exactly what he wants	RHG	BH/04 JS/RC08
EX43	For preventing problems, a lengthy preliminary stage is necessary, from which a detailed programme of demands should emerge	RHG	JS/RC02 JS/RC08
EX44	Landmarks need to be set and it has to be decided when certain deliverables need to be delivered	RHG	JS/RC15
EX45	The steering and consultation committees that keep an eye on things have to be expert	RHG	JS/PUBRC01 BH/03
EX46	It is necessary to draw up a proper risk analysis	RHG	JS/PUBRC05 EV/07
EX47	There was insufficient support in the organisation	RHG	JS/PUBRC01
EX48	Clients and project leaders should not just look at the result of the ICT project but also at how the organisation can be adjusted in order to achieve the higher goal by means of the project	PG	JS/RC01 JS/RC02
EX49	There is a gross lack of professionalism in the world of ICT. Only a very small section of people that are executing ICT projects at the moment have actually qualified in informatics. Surely, that is no good whatsoever!	CV	BH/01 PN/10
EX50	The government should really just work with accredited information scientists and not with self-educated people	CV	BH/01 PN/10
EX51	More money should also be invested in software research	CV	BH/01 PN/10
EX52	Universities should train people better in managing and executing large ICT projects	CV	BH/01 PN/10
EX53	Political deadlines are deadly as far as IT is concerned	CV	BH/02
EX54	If the elapsed time of an IT project is 'compressed', the costs will be excessively higher	CV	TAH/14
	Clinger Cohen Act (CCA) (1996): 8 minimal guidelines for standing a chance of funding!	CV	
EX55	Large investments should support the core tasks of the organisation (CCA 1)	CV	JS/PUBRC02 JS/RC01 JS/RC02
EX56	Only invest when there is no alternative, neither in the market, nor in another department (CCA 2)	CV	JS/RC01 PN/06
EX57	Before computerising, first optimise the work processes (CCA 3)	CV	JS/PUBRC02 JS/RC02
EX58	Prove that the expected Return On Investment for what you are about to do is better than alternative usage of public sources (CCA 4)	CV	JS/RC01 PN/06
EX59	The project should be consistent with the federal and departmental information architecture (CCA 5)	CV	JS/RC02 JS/RC19

	Risk reduction (CCA 6):		KY/03 EV/07
EX60	No isolated tailor-made solutions for parts of the project because this could work out very unfavourably for the whole project	CV	JS/RC01 JS/RC02 JS/RC06
EX61	Make use of fully tested pilots, simulations or prototypes before starting production	CV	JS/RC32
EX62	Clear-cut targets and accountability on the progress	CV	JS/PUBRC04 JS/RC02
EX63	Make sure that users are actually participating, help deciding and thinking along	CV	JS/PUBRC02
EX64	Investments should be realised in small bite size portions, in a relatively short period of time and each piece should by itself contribute to the core tasks and deliver a measurably positive result, which is independent of any possible pieces to be built later (CCA 7)	CV	JS/PUBRC03 TAH/14
EX65	Use an acquisition strategy that divides the risks between government and outsourcer reasonably and fairly (CCA 8)	CV	JS/PUBRC06 JS/RC09

Table 8.2: Recommendations/remarks of the experts as a result of the report produced by the Court of Audit

Advisors/Experts

Abbreviation	Name
CV	Chris Verhoef, Professor in Informatics at the VU
DR	Daan Rijsenbrij, chairman Platform Outsourcing Nederland
GS	Gerard Sanderink, CEO of ICT supplier Centric
HD	Hans Dieperink, General Management Inter Access
HN	Hans Nieuwenhuis, CIO Getronics PinkRoccade
JS	Jaap Schekkerman, founder of the online IFEAD think tank
JW	Johan van Wamelen, Professor at the Centrum voor Publieke Innovatie (Centre for Public Innovation), EUR
PG	Pieter Gremmen, Senior Project Manager Twynstra Gudde
RHG	Rob van den Hoven van Genderen, lecturer ICT law at the VU

Table 8.3: Overview Advisors/Experts.

8.4 SUMMARY

8.4.1 Introduction

In the tables 8.1 and 8.2, the items (success/failure factors according to the Court of Audit and remarks/recommendations from experts) have been linked to SUFFIs of the SUFFI Chart. In this, the main applicable SUFFIs have been shown. In some cases, a one on one relation applies, such as for example with regard to item AR3: "Political deadlines can be fatal to a project", which is related to SUFFI BH/02: "Deadlines are unrealistic".

In other cases, an item is related to a combination of SUFFIs. One example of this is item AR7: "Make sure to have a grip on your ICT projects", which is related to SUFFIs BH/01 ("Poor project management") and JS/RC15 ("Poor project planning, management and execution"). This meant that when the SUFFIs BH/01 and JS/RC15 are transformed from failure factors to success factors, AR7 would also become a success factor.

8.4.2 Conclusions

- in the tables 8.1 and 8.2 all items are related to one single SUFFI from the SUFFI Chart or to a combination of SUFFIs. On the basis of this, the SUFFI Chart does not need to be extended;
- nevertheless it is advisable to demand closer attention for:
  - **AR33/EX24**
    With regard to ICT, the government will have to make choices, in short will have to apply portfolio management;
  - **EX12**
    (hidden) political unwillingness for cooperation;
  - **EX13**
    For that reason, client and supplier bave insufficient elbow room within the execution for diverging from the original task, even though this would objectively be better for the client;
  - **EX49/EX52**
    There is a gross lack of professionalism in the world of ICT. Only a very small section of people that are executing ICT projects at the moment have actually qualified in informatics. Surely, that is no good whatsoever! Universities should better train people in managing and executing large ICT projects;
  - **EX59**
    The project should be consistent with the federal and departmental information architecture.

8.5 DEFINITION OF THE PROBLEM [Answer to sub question 7]

Definition of the problem, sub question 7:
Is it possible to apply the SUFFI Chart in the "Netherlands Court of Audit" case?

Answer:
Yes. In the tables 8.1 and 8.2, the items (success/failure factors according to the Court of Audit and

remarks/recommendations from experts) have been linked to SUFFIs of the SUFFI Chart. In this, the main applicable SUFFIs bave been shown. In some cases, a one on one relation applies, such as for example with regard to item AR3: "Political deadlines can be fatal to a project", which is related to SUFFI BH/02: "Deadlines are unrealistic".

In other cases, an item is related to a combination of SUFFIs. One example of this is item AR7: "Make sure to have a grip on your ICT projects", which is related to SUFFIs BH/01 ("Poor project management") and JS/RC15 ("Poor project planning, management and execution"). This meant that when the SUFFIs BH/01 and JS/RC15 are transformed from failure factors to success factors, AR7 would also become a success factor.

CHAPTER 9

DISCUSSION OF RESULTS

9.1 INTRODUCTION

This chapter describes the author's*) experiences. These experiences do not just relate to the projects and project audits from the portfolio but also to a few other projects and audits in which the author was involved and to observations on projects in which the author was not immediately involved but that did occur in environments where the author was active.
Personal experiences of the author also play a part in this chapter.

9.2 PORTFOLIO OF PROJECTS

The portfolio consists of 9 projects with external project-based publications and 4 (project-based) audits. Table 1.1 contains these projects and audits. About the projects the author has written 12 project-based publications in Dutch journals and 33 internal publications. The 9 projects represent an effort and duration of about 16 years. The author's role in projects 6-9 was: *internal* project manager at the Delft University of Technology. Regarding the projects 1-5, the author was the *external* project manager. The 9 projects on which external publications in the trade magazines appeared are discussed in sections 5.2 up to and including 5.10. In section 5.11, four (project-based) audits come up for discussion.

Chapter 5 and table 6.1 show that the 9 projects from the portfolio can be qualified as successful, although some projects during the project realisation period could be regarded as 'troubled projects' for some time (such as for example, the Telephony project). The first three project audits concern projects that had the status 'troubled project', which necessitated a project audit by an independent auditor in order to find out what was going on, as well as for making recommendations in order to get the projects back on track.

The 9 projects are all projects that were published on in the usual trade magazines. In addition, the author has realised a number of other projects over the last 43 years. These were not included in the portfolio. Not because these were not interesting for reasons of content and/or from a management point of view but simply for limiting the number of projects to be discussed. The 9 projects in question were chosen because publications play an important part when gaining a doctorate.

*) in this chapter: the author is AvD

Section 1.1 refers to research carried out by The American "Standish Group" [The Standish Group International 2003]. The author's experiences with projects within ICT over the last 43 years diverges from these research results. Not one of the author's projects was aborted prematurely. Some projects took more time/money than anticipated or were temporarily classed as 'troubled project' but practically all projects were successfully implemented. Only the results of two projects the author was involved with have not been put into use:

- the CCIP project for the FORTIS BANK Netherlands. This project did produce the required result: an application (SCOPUS) was procured by FORTIS and was adjusted for FORTIS to be used for the support of operational ITIL processes. In the week, that the project result was delivered it was announced that the supplier of the SCOPUS application had been bought up by SIEBEL. For the FORTIS management, this was reason to put the project result on ice till further notice;
- during the author's participation in a project for the Dutch Ministry of Defence there was a sudden order resulting in all external staff, through financial difficulties, having to leave the (sizeable) project at the end of the month in question. The author was in the middle of his activities and therefore not able to conclude his part in the project properly.

How is it possible that the author's dozens of projects of the past decades were concluded successfully, whilst so many other ICT projects tend to fail? In the opinion of the author, this is to do with a combination of factors.

The successful projects in question included factors related to SUFFIs from the SUFFI Chart:

- a very motivated project leader with great drive, stamina, perseverance and sense of responsibility (¬BH/01);
- a project leader with good abilities to find solutions (¬PN/04, PN/10, ¬EX6);
- a project leader who is strongly focussed on quality and who has extensive (information technical) expert skill and who in addition to the daily activities is involved in permanent education (PN/10, ¬EX49);
- a project leader who is expressly interested in embedding the project results in the user organisation and who involves (representatives of) the users in his project (¬JS/PUBRC02, ¬BH/05, EX63);
- a project leader who draws up realistic plans based on a Business Case, which have some flexibility, as well as monitoring these (PN/06);
- a project leader who enjoys proceeding methodically but uses the method as a tool and not as a target (JRR/02, TG/01, ¬PN/16);
- a project leader who, wherever possible, delivers the project results phased (¬JS/RC06, ¬JS/PUBRC03);

- a project leader who tries to communicate adequately with the different parties (¬BH/03, ¬KY/08, ¬NB/06).

The successful projects in question also included factors related to *new SUFFIs*:

- a project leader who has worked with small teams and who often acted as working foreman (AvD/01*);
- a project leader who also looks for projects that appeal to him, also from an application point of view. Preferably not too sizeable but challenging/difficult (AvD/02*);
- a project leader who rarely has problems with his project officers and manages to motivate them well (TAH/16, AvD/03*);
- a project leader who likes being given/takes the opportunity to round up a project completely, including documentation, evaluation and possibly with a publication in a trade magazine (AvD/04*).

*) These new SUFFIs AvD/01 and AvD/04 inclusive can be added to the SUFFI Chart and the SUFFI Total Chart.

In the next section, a number of the author's experiences will be discussed in more detail using Big Hitters. There is no claim to being exhaustive.

9.3 BIG HITTER 1: POOR PROJECT MANAGEMENT

All researchers as listed in table 4.2 mention "Poor project management" as a Big Hitter. Many books have been published on project management of ICT projects [including Gilb 1988, Dalcher&Brodie 2007]. Everybody is able to picture something when project management is mentioned but it is very difficult to give a univocal definition. A number of aspects are stated in 9.2. In this section, some aspects are discussed in more detail.

Business Case

For realising a project, it is important that everyone involved knows what target has to be achieved. The Business Case (BC) serves that purpose. The BC is the justification of the Business. In the author's experience, the project leader often assists in realisation of the BC. Of course, the BC needs to be realistic [JS/RC01, JS/PUBRC02]. The author stresses SUFFI PN/06: *The use of a business case results in a higher degree of satisfaction with the project* and TAH/16: *A mechanism for motivation, which is attracting interest in the software engineering field, is "goal setting".* Having a clear target within a project is not a new concept. Within social sciences, it has been a known fact that having a clear target in mind is an important success factor. Take for example the OKAPI project: in close consultation with a key user, the desired functionality of the new information system was determined after an initial study, using existing systems, documents and conversations. The delivery date was set by the Board of Governors of the UoA. On the 1st of July 1992, the existing computer was replaced by a different computer. The old information system could not run on the new computer. The BC was clear: "*delivery on 1 July 1992 at the very latest of an information system that did offer the desired (new) functionality*". The BC / target was clear both for the Business as well as the project team. Based on the BC, plans were drawn up and monitored. The project was split into 2 phases: phase 1 that included all necessary functions for being able to go into production and phase 2 that delivered the remaining functions. The author has used this working method, phased delivery, in many projects. The advantage is that the project team is initially able to focus on the truly important matters and can pass on the less important ones to phase 2. After having realised phase 1, phase 2 can be tackled. This working method also has another advantage: after delivery of phase 1, the user organisation has gained a better idea of the (desired) information system and the desired functionalities for phase 2 can be evaluated. Sometimes, this leads to changes or possibly (extreme) requirements can be cancelled through improved insight. However, this working method also has a disadvantage: sometimes Business managers are happy with the functionality of phase 1 and do no longer cooperate in phase 2 ("*that will have to do them, it has cost enough already!*"). The importance of the BC was also proved in the Kolihrie (sub) project (KPN Telecom). It was part of the Invoering BaanERP project. This was a sizeable project. The project team consisted of over 100 employees, of which the majority were brought in from outside the company. A good BC was made. It turned out that the costs of the entire project could be funded from the benefits of the project. These benefits came to millions of Euros and were realised because invoices no longer fell between two stools (forgot to make or forgot to send) and because all invoices could be sent on time (sometimes months sooner) to the customers.

Phased delivery

As stated under the subject Business Case, the author has in many projects strived for phased delivery in two phases. The applied method often was SDM, which is a waterfall method. Therefore, phased delivery is also very well possible when applying SDM. The opinion of Tom Gilb [1988], Jaap van Rees [1982] and Peter Noordam et al. [2007] is also applicable in this case. In new methods, such as Agile methods and EVO (Evolutionary Project Management), delivery in complete parts that can be used by the customer plays a very important part. In the Agile range of thoughts, the role of the team often takes central stage. Agile also means: more learner control, team responsibility, empowerment of the project officers. The project leader in an Agile environment focuses mainly on coaching and encouraging leadership. Project officers with insufficient seniority or lack of responsibility that cannot handle this freedom, cannot participate in an Agile environment [Louws 2008].

Different types of project leaders

The project leader acts as a spider in a web. He has to be capable of resolving problems because problems do occur in every project. Not every project leader is suitable for every project. Some projects demand a lot of technical knowledge. In that case, it is highly desirable that the project leader does have this technical knowledge (SUFFI PN/10). Project leaders that manage in a directive manner are out of place in Agile projects [Louws 2008]. In the author's opinion, experienced project leaders spend only about 10% of their time on planning and administrative activities and 90% of their time on project officers and activities with regard to the contents of the project.

Support

It is desirable that the project leader does have technical knowledge (SUFFI PN/10). If the project leader has insufficient technical knowledge, he is able to hire a technical project leader, provided that this is possible within the budgetary funds. In the opinion of the author, it is important to involve the ICT management organisation from the start of the project, especially in sizeable projects. By involving ICT Management at an early stage in the project, the requirements of ICT Management can be included at the right moment and problems at going into production of an application are prevented (see also section 9.5). However, many project leaders do not do this. They prefer not to have anybody looking over their shoulders and this method causes delays. However, they do forget that problems during the transfer to ICT Management can cause many delays and involve many costs. The same goes for IT Auditing. A sensible project leader includes calling in of an IT Auditor as an item in his budget.
The author thinks that it is better for a project leader to "quarrel" with ICT Management and the IT Auditor at the early stages of a project than at the end of the project. In various organisations, the author has succeeded in demonstrating that project leaders may consider an IT Auditor as a (strict) "friend" instead of an "enemy". Of course, the auditor stays in his role of auditor but at the start of the project for example, the auditor can pass a number of instructions to the project manager. He can also possibly communicate to the project manager (part of) the framework of norms on the basis of which the audits will take place. In an environment where one has to comply with SOX, the auditor may for example point out the SOX norms to the project manager ("show me" instead of "tell me"), thus enabling the project manager to take these into account. This makes more sense than waiting until the project has been completed and establishing that it did not comply with the SOX norms. In large projects, the project manager may also call in the support of a Financial Controller for monitoring the financial aspects.

Planning

The projects included realistic plans, even during the sixties and seventies. However, in those planning the environment was taken into account. For that reason, the necessary room was introduced in certain projects. SUFFI TAH/57 reads: *The "Safety Factor Policy" does achieve its intended objective: more accurate estimates. However, the organisation pays dearly for this (see table 9.3.2).* In theory, the author does agree with this conclusion. Nevertheless, he did include extra room in the planning in some projects. The reason being that in the environments in question no actual result could be achieved by means of acting in a directive manner. The project leader had to try to find his way within the organisation tactfully and try to convince the staff of the organisation into doing their best for his project. This calls for attention and therefore will cost the employees in question time. When working with a tight schedule in environments like that, this will result in stress situations that act counterproductive. The delays that may result from that are far worse than the extra room in the planning. All this does not apply to any environment. SUFFI TAH/80 is also applicable here: *A different schedule creates a different project.*

The project planning can be supported hy a software package. Various software suppliers supply adequate software. It is important not just to measure the quantity of used resources hut also the quantity of the achieved results. In paragraph 3.8.3 SUFFI TAH/11 came up for discussion:

TAH/11: It is difficult to measure performance in programming {181}

Remark (AvD):
TAH/34: Progress, especially in the earlier phases of software development, is measured by the rate of expenditure of resources rather than by some count of accomplishments {81}.

This often occurs but can lead to disappointment in the further course of the project. Both the used resources as well as the progress (results) need to be measured. The figure below shows that the used resources and results are in balance at a vertical line. This is the case with line 2 at point in time t2. In that case, for example, 35% of the available resources have been used and 35% of the required results have been achieved. Line 1 at point in time t1 indicates that the results are running behind with the used resources at point in time t1. Line 3 at point in time t3 shows that the results are ahead of the used resources.

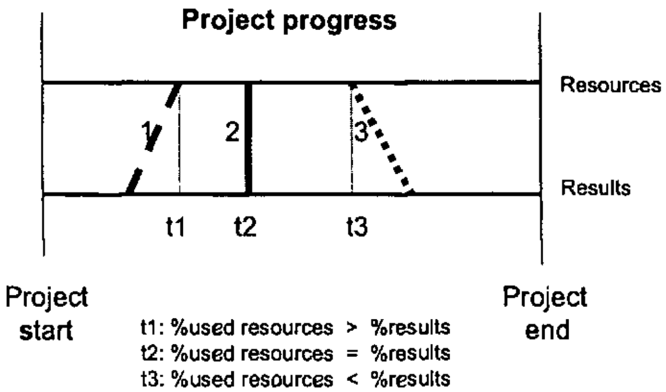


Figure 9.3.1(=3.8.1): Project progress

"It is difficult to measure performance in programming {181}". This is often the case, although there are of course possibilities for measuring the progress. For instance, the OKAPI project (business case in section 5.6). The online part of OKAPI consists of 132 display images that are divided over five sub systems. The figure below is a (schematic) example of a display image. The coherence of the 132 display images was shown during the construction in so-called display image communication diagrams. The large drawings were hanging on the wall. The four corners of a display image were used for showing the progress:

- *corner1 was coloured green as soon as the program in question (dialogue stated) was coded;*
- *corner 2 was coloured blue as soon as the dialogue in question was tested;*
- *corner 3 was coloured yellow as soon as the dialogue in question was tested in coherence with the surrounding dialogues (integration test);*
- *corner 4 was coloured red as soon as the acceptance test by the key user in question had taken place successfully.*

That way any interested employee/ manager was able to follow the progress by looking at the wall of the project room. Providing the dialogues with a weighting factor in advance made it possible to establish the percentage of the progress using a simple calculation programme. It was important that an evaluation with regard to the weighting factor took place after realisation of a dialogue. Was this correct? If not, then what were the consequences for other dialogues yet to be realised? Changes to the weighting factor of yet to be realised dialogues sometimes did have consequences for the planning and progress. Over the course of the project, the changes with regard to weighting factors became less through advancing insight. The wall overview with or without (partly) coloured display images did present a clear idea of the progress of the project and was highly appreciated by the management.

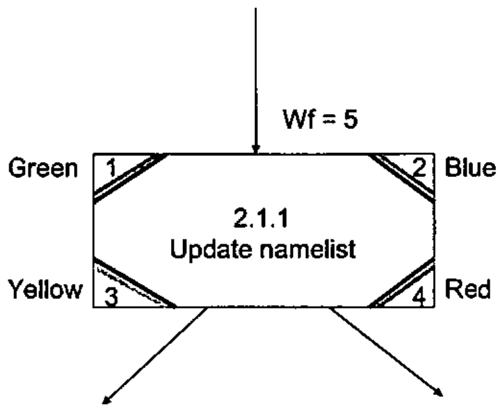


Figure 9.3.2(=3.8.2): Measuring the progress of the OKAPI project

Table 9.3.1: Measure performance in programming

SUFFI TAH/57 reads: *The "Safety Factor Policy" does achieve its intended objective: more accurate estimates. However, the organisation pays dearly for this.*

Safety Factor Policy

	Method A	Method B	Method C
Safety factor	0%	50%	150%
Mandays estimated	2,359**)	3,538	5,900
Mandays actual	3,795	5,080	5,412
% Error (relative)	+ 38%*)	+30%	- 9%

*) $\frac{3{,}795 - 2{,}359}{3{,}795} = +0.38$

**) Project: 64 KDSI (Delivered Source Instr.) (Boehm/COCOMO, 1981)

The "Safety Factor Policy" does achieve its intended objective:
More accurate estimates.
However, the organisation pays dearly for this because the project consumes 34% resp. 43% more mandays!!!

TAH: A different estimate creates a different project!!!

The above example was taken from the book by Abdel-Hamid and Madnick [Abdel-Hamid & Madnick 1991]. What is the influence of the "Safety Factor" on the ultimately required effort in man-days and therefore on the correctness of the estimating and the costs? The example concerns a project of 64 KDSI (DSI = Delivered Source Instructions). Using Boehm's COCOMO method (COCOMO = COnstructive COst MOdel) it was calculated that the estimated required number of man-days for executing the project is 2,359 days. The Abdel-Hamid model shows that at a 0% Safety Factor, the truly required number of days will be 3,795 days. A relative error of +38%. In the case of a 50% Safety Factor, the relative error is +30% and at a 150% Safety Factor, the relative error is -9%. All this means that the relative error of the estimated number of man-days decreases at a higher Safety Factor. However, the actual number of days and therefore also the costs of the project increase. The following question has to be asked: "Do we want the best estimation or do we want the lowest costs?". In this example, the 150% Safety Factor results in the best estimation but also in the highest costs. It has to be noted that in this example other aspects, such as the quality of the end products and the working method within the project, have been left aside.

Table 9.3.2: The Safety Factor Policy

Level of training

In table 9.3.3 SUFFI JS/RC23 is mentioned. Suppliers do not always put in the most suitable employees but often those workers that are currently available. That does not always result in a good match. Several experts have stated (see table 9.3.3) that the level of training is often below the desired level, as Professor Verhoef for example says in EX49: "*There is a gross lack of professionalism in the world of ICT. Only a very small section of people that are executing ICT projects at the moment have actually qualified in informatics. Surely, that is no good whatsoever!" and* in EX50: *"The government should really just work with accredited information scientists and not with self-educated people"*.

The author can confirm this. Even stronger, he has missed out on an order several times because the management in question preferred to not have a project leader/employee with a lot of knowledge and experience on the team because the management was afraid that it would fail. We don't want anyone looking over our shoulder!

JS/RC23	Vendor lack/loss of skilled resources
EX6	*Failing project management:* Through a lack of knowledge and experience and insufficient problem-solving capacity, small problems develop into major problems
EX7	*Failing IT expertise*: Through a lack in technical knowledge as regards to content, there is too much dependency on externals that often have their own agenda surrounding the projects
EX49	There is a gross lack of professionalism in the world of ICT. Only a very small section of people that are executing ICT projects at the moment have actually qualified in informatics. Surely, that is no good whatsoever!
EX50	The government should really just work with accredited information scientists and not with self-educated people
EX52	Universities should train people better in managing and executing large ICT projects

Table 9.3.3: Level of training/knowledge

In section 6.2 I have stated that "Lack of senior management involvement and commitment" can be considered as Big Hitter 6. In my opinion based on the above, the contents of table 9.3.3 and a number of TAH SUFFIs such as TAH/03, TAH/05 and TAH/09, "Lack of professionalism" can be viewed as Big Hitter 7.

9.4 BIG HITTER 2: DEADLINES ARE UNREALISTIC

As early as 1982, Professor Oonincx published his book "Waarom falen informatiesystemen nog steeds?" (Why are information systems still failing?) [Oonincx 1982]. His main conclusions were (see table 9.4.1): *"Information systems, which are set up too ambitiously, too isolated or without proper* *planning, stand a very large chance of failing. Insufficient involvement of future users in the development of information systems or a passive attitude of the top management also often lead to disappointing results".*

The author and many others with him, think that no serious planning is made in similar situations but the client sets a deadline that is not founded. In 1982 as well as today, that led to failing IT projects.

Although there are many well-functioning information systems, there are plenty of information systems that either function badly or never reach their original target as set during construction, namely being taken into service. This book discusses, grouped in seventeen chapters, a number of aspects that often cause development and introduction of information systems to fail. Information systems, which are set up too ambitiously, too isolated or without proper planning, stand a very large chance of failing. Insufficient involvement of future users in the development of information systems or a passive attitude of the top management also often lead to disappointing results. The book is suitable for students of information science, business administration, economics etc. Although the IT engineer, employed in practice will encounter a number of familiar cases, he will also benefit from reading the book attentively. (Future) users of information systems will largely reap the benefits from studying this book. In this case, the argument "no time" will not wash. It is a very readable book of modest proportions but with very important contents.   *25 August 1982 - ir. Aart J. van Dijk*

*Table 9.4.1: Book review of the book of Professor Ooninc*x

At the end of November 2007, the Netherlands Court of Audit published the report "Lessen uit ICT-projecten bij de overheid – Deel A" [Wijsman et al. 2007] (Lessons from government ICT projects – Part A). Various experts have commented on this report (see chapter 8).

Table 8.1 includes the conclusions of the Court of Audit represented by 39 success/failure factors. In table 8.2, the recommendations/remarks of experts have been recorded in the shape of 65 items of recommendations/remarks. For all 104 items, it was checked which SUFFIs from the SUFFI-Chart can be related to the item in question.

Conclusion

As far as ICT projects are concerned, the government handles these badly. The Court of Audit arrived at this conclusion in their research report [Wijsman et al. 2007].

Table 9.4.2. includes a few items from tables 8.1 and 8.2, such as AR3: *Political deadlines may be fatal to a project* and EX2: *Political inability: Unclear processes, vague agreements and compromises cannot be computerised.* SUFFIs JS/RC01 and JS/RC07 are also included.

JS/RC01	Project based on an unsound premise or an unrealistic business case
JS/RC07	Vendor setting unrealistic expectations on cost, time-scale or vendor capability
AR2	ICT is not a 'quick fix' for a problem. The decision-makers in the political field believe ICT to be a miracle cure for solving all sorts of policy issues
AR3	Political deadlines may be fatal to a project
AR11	Government ICT projects are often too complex
AR18	In complex projects there is no balance between ambition, available manpower, means and time
AR38	A project can be made manageable by dividing it into smaller sub projects that are easier to manage
EX2	*Political inability*: Unclear processes, vague agreements and compromises cannot be computerised
EX8	*Opportunism in IT companies*: IT suppliers are too eager to build solutions that would really be advised against because the customer is not yet ready
EX9	*Opportunism in IT companies:* Lesson one is to reorganise first and only then to computerise
EX31	In the case that the ambitions of politicians cannot be restricted, the commissioned projects will have to be divided into bite size modules. The principles of architecture may offer guidance when modulising
EX33	Monitor the scope of projects, avoid 'scope creep'
EX39	It is of major importance to have independent experts performs audits on a regular basis, during the execution as well. This is a task for the Court of Audit
EX46	It is necessary to draw up a proper risk analysis
EX50	The government should really just work with accredited information scientists and not with self-educated people
EX59	The project should be consistent with the federal and departmental information architecture
EX64	Investments should be realised in small bite size portions, in a relatively short period of time and each piece should by itself contribute to the core tasks and deliver a measurably positive result, which is independent of any possible pieces to be built later

Table 9.4.2: A few aspects concerning unrealistic deadlines

The report of the Court of Audit shows that: *Political deadlines may be fatal to a project (AR3).* The report as well as other studies (see Chapter 3) demonstrate that the causes as mentioned by Professor Oonincx in 1982, are largely applicable today. Did the report of the Court of Audit already result in a revolution within national government?

That is highly doubtful. Table 9.4.3 includes a newspaper article daring from 7 November 2008. In the opinion of the author, politicians still have not drawn visible conclusions from the Court of Audit report and there is every chance of a new project failure.

An interesting exception is the performance of Minister Eurlings in the Dutch TV programme "Buitenhof" in May 2009. In this programme, the Minister stated that the project "introducing the kilometre tax" (table 9.4.3) will not be finished in 2011 but no earlier than 2013. He also declared that "from today priority will be given to a properly controllable project and not to desired planning". Thus ended an unrealistic planning. There was ample criticism from politicians because he was unable to honour the promises he made but there was also sympathy for his viewpoint deciding to allow the rest of the project to run controllably. A fault confessed is half redressed!

'Time bomb' under kilometre tax plan

The planning for introducing the kilometre tax is that tight that the project threatens to fail. Experts even speak of a time bomb. The term time bomb was mentioned in an analysis of the Ministry of Transport, which quotes a confidential report on the introduction of road-pricing.

Minister Eurlings still assumes that lorries will be the first to start paying kilometre tax from 2011. A year later, they will be followed by the first private cars. The Minister has already declared that the chance of a delay is 85 percent. Only 'if all goes well' we will be able to make 2011, Eurling's spokesperson said yesterday. However, in order to put the pressure on, the Minister wishes to hold on to this date for the time being.

However, Eurling's own 'departmental audit service' reports that several expert reports state that the planning is unrealistic. At least a year's delay seems inevitable. According to Eurling's civil servant, the gap between the 'if all goes well planning' and the realistic planning has by now become that wide that this presents 'a risk to adequate management of the project'.

Mark van der Werf in Dutch daily Algemeen Dagblad of 7 November 2008

Table 9.4.3: 'Time bomb' under kilometre tax plan

The author's projects

As far as the author's projects are concerned, all projects started based on a realistic planning with some room. Proper measuring meant that a finger was kept on the pulse and where necessary it was possible to adjust. All plans came into being after consideration with the clients. Of course that did result is some heated discussion but all those concerned always managed to come to agreement on the planning. Choosing for phasing many projects into two phases meant an important positive contribution in this process.

Over the past 43 years, it happened twice that the views on an (achievable) planning were that far apart that a (compromise) solution was not found. In those cases, the author decided it to be preferable not to participate in the projects in question. Those projects were not successful.

9.5 BIG HITTER 3: POOR COMMUNICATION

Communication within and surrounding the project is not just very important but also proves to be difficult. In Chapter 4, four researchers have stated its importance (see table 9.5.1).

KY	08	Poor internal communication
LM	08	Communication breakdowns
NB	06	In cases of failure there is too little appreciation and attention to the quality of the (written) communication
PN	02	Lack of communication in and around the project (KPMG)
LM	10	Late failure warning signals
NB	05	Problems escalated too late

Table 9.5.1: Big Hitter 3 - Poor communication and late warning signals

Telephony (DUT)

Poor communication can lead to problems being noticed too late. This was the case with the DUT Telephony project. Within the project team, PM-I&I caused bad communication, which resulted in a "troubled project". For that reason, the author (PM-B&B) and PM-FU were asked by the management of the Computer Centre to act as crisis managers. Only thanks to the all-out effort of those involved working long days and the excellent assistance of supplier Lacis, the project was delivered on time (see paragraph 5.5.3).

AUBID

The AUBID project concerns the computerisation of the Library of the Delft University of Technology (DUT) in 1979. The communication with the Library management was very difficult. The Managing Director did not really want a computerised information system. The Assistant Manager really wanted the Library to have its own computer. However, the Board of Directors had decided to buy a foreign library system via IBM, which was to run on the central computer of the Computer Centre (see paragraph 5.8.3). Over the course of the entire project, any communication with the management was very difficult, which caused delays. Thanks to the all-out effort made by the middle management of the Library and the project team, the AUBID information system was delivered on time.

PQRS)*

PQRS is an insurance company in the east of the Netherlands. In 1995, the author was asked to carry out an IT Audit at the PQRS Computer Centre. People complained about the performance of the AS 400 computer and the management wished to find out what was wrong. After an investigation of a week, which included a number of interviews, the author met the head of the Computer Centre on

Friday at 4 pm for a final discussion. One of the author's observations was that the computer was currently (that particular week) charged too heavily for realising a good performance. The defence of the Head was that other departments, including Application building, made little or no plans and he was regularly confronted with a fait accompli. The author was also given that impression but in his opinion, the Head could also act a little more assertively. During the meeting, around 4.30, the Head of Applications entered the room. He handed the Head of the Computer Centre a few programmes and documentation saying: "We have built and tested this information system and it has to run in production next Monday because all Business departments will have to work with it!". The Head of the Computer Centre was rather surprised and "not amused" because the size of the information system was such that it could not be simply added to the AS 400. The AS 400 had to be extended first. The author included this case study in his audit report for illustrating the "lack in on time and adequate communication". The Business departments were also "not amused".

*) This name has been invented. The case itself is truthful.

9.6 BIG HITTER 4: INCOMPLETE /WEAK DEFINITION REQUIREMENTS

Drawing up of the requirements that a new information system has to meet is not all that easy. Within SDM this is realised in the "Definition study". Based on an approved definition study, a functional design is made. Next, the information system is built. Requirements can be drawn up at a global level but if necessary can also be very detailed. In some environments, users are invited to draw up requirements. To that purpose, a user group is created. The author's experience with such user groups is diverse. Sometimes these function well but often one does not succeed in drawing up a good consistent collection of requirements. Such user groups often require supervision. The author has positive experiences with regard to drawing up requirements in collaboration with some representatives of users. In that case, these representatives of users do need to know what's what and also need to give feedback to their grassroots. Prototyping may help when drawing up requirements. A few examples.

AUBID

The AUBID project concerns the computerisation of the Library of the Delft University of Technology (DUT) in 1979. In the project team, some of the library staff participated (see also 9.7). An important business unit within the library is the "Lending" department. The requirements of this department have been drawn up after consultation with the department staff and two officers of the project team. On several occasions, the author and a project officer also spent some time in this department in order to observe the activities "in full flight". The requirements were developed by the project team and submitted to the business. All this worked out fine. The delivered information system met all requirements and the users were very happy.

GEMS

For the benefit of all the DUT warehouses, the eighties saw the development of an advanced information system for warehouse administration and management [Van Dijk et al. 1983]. One of the most high-profile warehouse managers was able to disengage himself part time in order to participate in the project team. It took a lot of time to draw up the detailed requirements. During the building, the person involved regularly tested the results. One of the main advantages at the time being that when GEMS was delivered, there was no more discussion regarding requirements and functionality.

BRIDGE BUILDING

When working as a system designer/programmer for the Ministry of Transport and Public Works, the author was assigned to make a programme that was to support the building of a new bridge. The client, a civil engineer, had made the programme schedule himself and provided the (partly statistic) formulas. The author was unfamiliar with the subject matter. Nevertheless, using literature and common sense he managed to find a few errors in the formulas. The client was happy that the errors in the requirements had been detected at an early stage.

FINAD)*

The financial administration was computerised anew by a then well-known Dutch ICT company (since then taken over by another ICT company). The author was not involved in this but worked on a different project in the immediate vicinity. The requirements and first draft were informally communicated to the author by employees of the client company. Having looked at these documents broadly, the author was sure that the information system would never work properly. During a conversation at the coffee machine, the author asked the project leader in question about this. He answered that he knew that as well but they were building according to the requirements. If it proved not to work, then this would mean a new project, wouldn't it?!.

*) This name has been invented. Tbe case itself is truthful.

In the opinion of the author, copying delivered requirements unquestioningly entails great risks for the project in question.

AGILE/EVO

In brief, the Agile method means that a project is delivered at agreed times in small complete parts that are of use to the customer. The advantage of delivery in parts is that you will sooner receive feedback that can be used for the next parts. Of course, the information system has to be suited to be delivered in small complete parts.

Evo (Evolutionary project management / delivery) supported the related concept that projects do not initially have the 'final and correct user requirements' specified. The underlying principle of Evo is the Plan-Do-Study-Act cycle (see figure 9.6.1). The new system is delivered in a series of small steps.

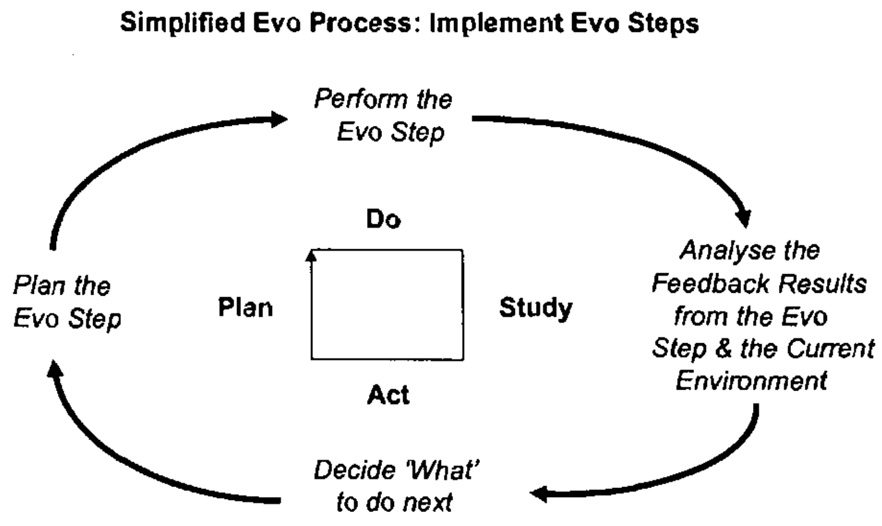


Figure 9.6.1: A simplified Evo process: Implementing Evo Steps [Gilb & Brodie 2005]

Figure 9.6.1 shows the Evo steps of the Plan-Do-Study-Act cycle [Gilb & Brodie 2005]:

Plan: 1. Specify the delivery of the step in detail.
2. Agree the plan with the relevant stakeholders.

Do: 1. Deliver the step.
2. Install it with real stakeholders, so they get some of the planned measurable benefits.

Study: 1. Determine the results of delivering the step. Compare results to the short-term and long-term targets.
2. Analyse the data and produce a feedback report for management.

Act: 1. Decide if this step succeeded, must be redone in whole or part, or totally rejected.
2. Take any required minor corrective actions to 'stabilise' the system.

Big Hitter 4 is: INCOMPLETE /WEAK DEFINITION REQUIREMENTS. Nevertheless, one characteristic of Agile/Evo projects is that at the start of the project, the requirements do not need to be known entirely and in detail. Furthermore, these can change. How is that possible? As said, the software is delivered in small complete parts (iterations). The users can set to work with these immediately and the developers will soon receive feedback. The users determine the priority of the iterations. Only when an iteration is started, the detailed requirements have to be known. When changes occur, new iterations can (partly) replace earlier iterations. This enables starting with a project whilst not all requirements are known at detailed level. It has to be noted that a quick short term solution may stand in the way of a future-proof solution. Furthermore, the future-proof solution will most likely cost more than a solution without the temporary interim solutions. There could be an impression that it is not necessary to plan for Agile/Evo projects because the planning will have to be constantly adjusted to the changing situation anyway or that it may not be necessary to work on the basis of an architecture. Nothing can be further from the truth. In the Agile working method, it is of vital importance for everything delivered to be of extremely high quality. When you are constantly iterating, you want to be able to trust everything that is finished as much as possible. The products of an Agile/Evo project will also be taken into production, so there is indeed a need for various types of documentation. In Agile projects, the role of the team often takes central stage. A directive way of management is not called for. As a project manager you focus on coaching and encouraging leadership. Project officers with insufficient personal seniority or lack of responsibility, cannot be used in an Agile environment.

9.7 BIG HITTER 5: INSUFFICIENT INVOLVEMENT OF FUTURE USERS

In the opinion of the author, involving future users in the development and realisation of information systems is very important and affects the quality (usability) of the information system. The author has given a lot of attention to involving users in his projects. Professional literature includes a few classic drawings that illustrate what the user intended and what he ended up with [Oonincx 1982]. Next, a few examples are discussed in more detail.

AUBID

The AUBID project concerns the computerisation of the Library of the Delft University of Technology (DUT) in 1979. The project group consisted of 5 members of staff, including two Library staff. The management of the departments involved (middle management) was very motivated and cooperated with the project team in a very constructive way. On the 31st of October 1979, the intermediary request system was officially put into use [see photos jp11]. The information system met all requirements. Both the library staff as well as the customers were very satisfied.

OKAPI

In the financial building administration of the University of Amsterdam the construction of new buildings and renovation projects are registered. The administration is supported by OKAPI. During the construction one of the objectives were: "The new information system should be available on the 1st July 1992 because at that moment the old information system is no longer available".

The key user is expressly involved in the project. The author did not just make a functional design but also a prototype. By means of this prototype, the users were able to get a good idea of the future system. The key user was very critical. Not just because he had already worked in the financial building administration for 25 years and therefore had in-depth knowledge of the subject matter but also because the existing information system that was running on the CDC, had been produced by him. It was not a professional system and he was the only person able to work it. Nevertheless, the author had greatly appreciated the initiatives that he had taken. Although he was no information scientist, he still managed to make a system that worked for him. The presentation in the last week of June 1992 of the new information system to the management of the University Office and the users was taken care of by the project leader in close cooperation with the key user. The information system was greatly appreciated by everybody.

GEMS

For the benefit of all the DUT warehouses, the eighties saw the development of an advanced information system for warehouse administration and management [Van Dijk et al. 1983]. One of the most high-profile warehouse managers was able to disengage himself part time in order to participate in the project team. He had major functional influence on the results and considered the information system as something he had built himself. One of the main advantages was that he successfully introduced the information system to the other warehouses.

FINAD)*

The financial administration was computerised anew by a then well-known Dutch ICT company (since then taken over by another ICT company). The requirements and first draft were informally communicated to the author by employees of the client company. Having looked at these documents broadly, the author was sure that the information system would never work properly. During a conversation at the coffee machine, the author asked the project leader in question about this. He answered that he did know that as well but they were building according to the requirements and further contact with the employees was not needed. After delivery of the information system it proved not to work properly.

*) This name has been invented. The case itself is truthful.

Prince2

The use of a structured project management method such as Prince2 (the name Prince stands for PRojects IN Controlled Environments) can promote involvement of the future users in the project. Prince 2 calls the reasons why a project needs to be started the Business Case. The Project Board is the steering committee of the project. These will asses the Business Case and the project risks before the start of each stage of the project by means of reports. Within Prince2 the Project Board is project

owner. In the Project Board, three roles are represented: *Business Executive*, *User* and *Supplier*. The project result has to contribute to the organisation's objectives, it is intended for the users and it is realised hy people using the resources supplied by the provider [Hendriks et al. 1997].
Prince 2 approaches project management process oriented. Characteristic of a process approach is the central place as taken by the objective of the process. The Project Board has to give a Go or a No Go hefore the start of each stage. The users are represented in the Project Board. In this fashion, the interests and the involvement of the users in the project can he guaranteed.
With respect to this, it is of course important for Prince2 to be applied in the correct, intended manner. It is for example necessary to assess at each new stage whether the Business Case is still valid. It is also necessary to give explicit permission for the next stage. This also allows the Project Board to check whether the future users have participated sufficiently and in the correct manner. In practice, Prince2 is often only used to a limited degree. In that case, one sometimes refers to Prince In Name Only (PINO).

9.8 BIG HITTERS: CONCLUSION

Looking at the ahove sections, one can establish that the derived five Big Hitters in Chapter 6 are in my opinion important SUFFIs as well. In section 6.2 I have stated that "Lack of senior management involvement and commitment" can he considered as Big Hitter 6. Based on what is stated in section 9.3 "Lack of professionalism" can be viewed as Big Hitter 7.
The SUFFI Chart and the SUFFI Total Chart can be extended with Big Hitters 6 and 7.

CHAPTER 10

CONCLUSIONS AND RECOMMENDATIONS

10.1 INTRODUCTION

One may ask the question, whether it is relevant to look at success and failure factors of ICT projects. It may be concluded that the subject of success and failure factors in ICT projects has been in the spotlight for more than 26 years and still is very topical (section 1.1).
In this thesis, the following problem definition is considered:

"How were the ICT projects the author) worked on (the portfolio of projects) managed (the key here is the author's observations and experiences) with regard to success and failure factors, and how do they agree or disagree with what the procedures in Tarek Abdel-Hamid's work on Software Project Management and others say happens with regard to success and failure factors?".*

Based on the problem definition, seven sub questions were defined. The sub questions were answered in chapters 2 - 8 of this thesis. In sections 10.2 and 10.3, the answers are summarised in brief. Section 10.4 (further research) indicates how the SUFFI Charts as constructed in this thesis can be further improved. In section 10.5 some recommendations are presented.

*) in this chapter: the author is AvD

10.2 SUB QUESTIONS

10.2.1 Sub question 1

What is understood by success/failure factors in ICT (for short: SUFFIs = SUccess and Failure Factors in ICT projects)?

Answer:

A project failure was defined in paragraph 2.3.2. For this thesis a project failure has one or more of the following characteristics:

- it does not comply with the functionality agreed to in advance, including agreed changes of scope;
- it exceeds the planned time-scale by more than 50%, excluding the time-scale impact of agreed changes in scope;

- it exceeds the build cost by more than 50%, excluding the cost of agreed changes in scope.

A success factor in ICT is a factor that contributes to the successful realisation of an ICT project, for example "Good project management" or "Excellent definition of requirements". A failure factor in ICT is a factor that contributes to a "project failure", for instance "poor project management (BH/01)", "incomplete/weak definition of requirements (BH/04)" or "premature declarations of victory (EV/04)".

10.2.2 Sub question 2

Is it possible to derive SUFFIs from international publications, Dutch publications and from the procedures in Tarek Abdel-Hamid's work on Software Project Management and if so, what particular SUFFIs?

Answer:

Together with tables 3.7.11, 3.7.12 and 3.8.2, table 4.2 forms the *"Reference model success and failure factors ICT projects" (for short: SUFFI model, SUFFI = SUccess/Failure Factors in ICT projects) (see figure 10.1).* That establishes "The opinion of others about SUFFIs in Software Project Management".

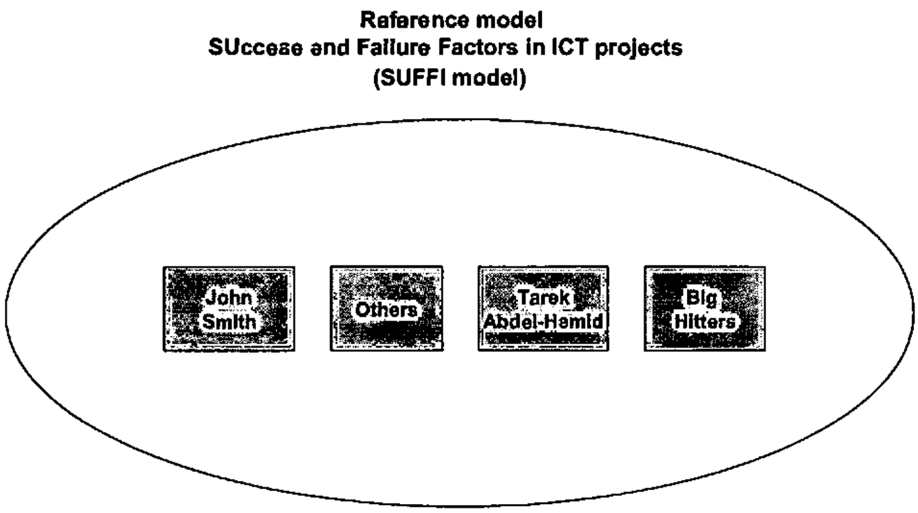


Figure 10.1(=4.1): Reference model SUccess and Failure Factors in ICT projects (SUFFI model)

10.2.3 Sub question 3

Are there any Big Hitters amongst the SUFFIs and if so what are these?

Answer:

Table 4.2 includes the success/failure factors that have been named as "Big Hitter".

The Big Hitters are:

- poor project management;
- deadlines are unrealistic;
- poor communication;
- incomplete/weak definition requirements;
- insufficient involvement of future users.

10.2.4 Sub question 4

Which SUFFIs are applicable to what particular project from the portfolio of the author's projects?

The results of the study are recorded in:

- table 5.2.1: Success/failure factors POTVIS project;
- table 5.3.1: Success /failure factors Kolibrie project;
- table 5.4.2: Success/failure factors Charging method project;
- table 5.5.1: Success/failure factors Telephony project;
- table 5.6.1: Success /failure factors OKAPI project;
- table 5.7.1: Success /failure factors GIRAF project;
- table 5.8.1: Success/failure factors AUBID project;
- table 5.9.1: Success/failure factors VDV project;
- table 5.10.1: Success/failure factors BIBLIOSYSTEM project;
- table 5.11.1.1: Success/failure factors NUMIS-2000 project;
- table 5.11.2.1: Success/failure factors SYSA project;
- table 5.11.3.1: Success/failure factors ACCINT project;
- table 5.11.4.1: Success/failure factors FINANCE projects.

10.2.5 Sub question 5

Is it possible to use the Big Hitters for distinguishing between successful and not successful projects from the portfolio of the author's projects?

Results cases	Apply to					Score		
	Big Hitter 1	Big Hitter 2	Big Hitter 3	Big Hitter 4	Big Hitter 5	Funct.	On time	Within Budget
Case 1: POTVIS project (KLPD)	No	No	No	No	No	Yes	Yes	Yes
Case 2: Kolibrie project (KPN Telecom)	No	No	No	No	No	Yes	Yes	Yes
Case 3: Charging method project (GAK)	No	No	No	No	No	Yes	Yes	Yes
Case 4: Telephony project (DUT)	No	No	No	Yes	No	Yes	Yes	No
Case 5: OKAPI project (UoA)	No	No	No	No	No	Yes	Yes	---#
Case 6: GIRAF project (DUT)	No	No	No	No	No	Yes	Yes	---#
Case 7: AUBID project (DUT)	No	No	No	No	No	Yes	Yes	---#
Case 8: VDV project (DUT)	No	No	No	No	No	Yes	Yes	---#
Case 9: BIBLIOSYSTEM project (DUT)	No	No	No	No	No	Yes	Yes	---#
--	-------	-------	-------	-------	-------	-------	------	--------
Case 10: Audit Multihouse	Yes	Yes	Yes	Yes	---+	No	No	No
Case 11: Audit SYSA (GOVERN)	Yes	Yes	Yes	Yes	Yes	No	No	No
Case 12: Audit ACCINT (PUBLIC)	Yes	Yes	Yes	Yes	Yes	No	No	No
Case 13: Audit SOX (FINANCE)	No	No	Yes	No	No	Yes	Y/N*	Y/N*

+) unknown
#) no specific budget available
*) Yes or No, depends on the project

Table 10.1(=6.1): Big Hitters in relation with the discussed cases

Answer:

Table 10.1 (see also chapter 6) contains the Big Hitters in relation with the discussed cases. Although the results are based on a very limited spot check, the conclusion may be drawn that the collection Big Hitters within this collection of cases acts discriminating. Where at least four of the five Big Hitters are not applicable, the "score" is positive. Where at least four of the five Big Hitters are applicable, the "score" is negative. This picture also corresponds with the author's experiences and observations during other projects and audits. When the five Big Hitters lead to a negative score, a large number of other SUFFIs usually play a part.

10.2.6 Sub question 6

Is it possible to present the SUFFIs in a SUFFI Chart that is easy to use in practice by others?

Answer:

The **SUFFI model** was constructed. It consists of tables 3.7.11, 3.7.12, 3.8.2 and 4.2. The SUFFI model can be put to use immediately and next further tested and improved. In order to simplify its use even more, the tables have been combined into one single table, the so-called **SUFFI Chart** (see figure 10.2). This is included in section 7.2.

The SUFFI Chart was created from a number of tables (3.7.11, 3.7.12, 3.8.2 and 4.2 and the Big Hitters 6 and 7 (see section 9.8)). These tables include the research results of various researchers. In order to be able to test new research results more easily against all tables, a **SUFFI Total Chart** was created in which all tables (3.7.1-3.7.10, 3.8.1 and 4.2 and the Big Hitters 6 and 7) were included. This is included in section 7.3.

SUccess and Failure Factors in ICT projects (SUFFI Chart)

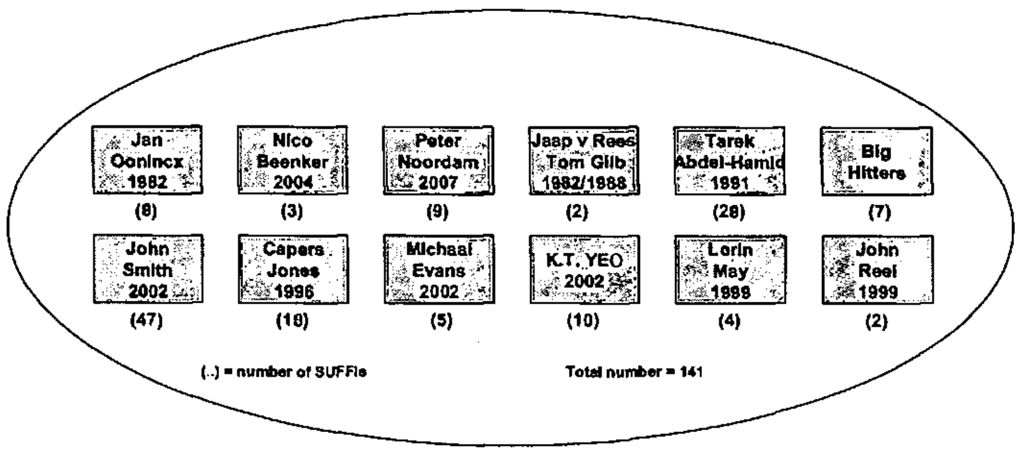


Figure 10.2: SUFFI Chart

10.2.7 Sub question 7

Is it possible to apply the SUFFI Chart in the "Netherlands Court of Audit" case?

Answer:

Yes. In the tables 8.1 and 8.2, the items (success/failure factors according to the Court of Audit and remarks/recommendations from experts) have been linked to SUFFIs of the SUFFI Chart. In this, the main applicable SUFFIs have been shown. In some cases, a one on one relation applies, such as for example with regard to item AR3: "Political deadlines can be fatal to a project", which is related to SUFFI BH/02: "Deadlines are unrealistic".

In other cases, an item is related to a combination of SUFFIs. One example of this is item AR7: "Make sure to have a grip on your ICT projects", which is related to SUFFIs BH/01 ("Poor project management") and JS/RC15 ("Poor project planning, management and execution"). This meant that wben the SUFFIs BH/01 and JS/RC15 are transformed from failure factors to success factors, AR7 would also become a success factor.

10.3 CONCLUSIONS

The answer to the research question (definition of the problem):

"How were the ICT projects the author worked on (the portfolio of projects) managed (the key here is the author's observations and experiences) with regard to success and failure factors, and how do they agree or disagree with what the procedures in Tarek Abdel-Hamid's work on Software Project Management and others say happens with regard to success and failure factors?"

is given in the paragraphs 10.2.1 – 10.2.7:

- For this thesis a project failure could defined and has one or more of the following characteristics:
    - it does not comply with the functionality agreed to in advance, including agreed changes of scope;
    - it exceeds the planned time-scale by more than 50%, excluding the time-scale impact of agreed changes in scope;
    - it exceeds the build cost by more than 50%, excluding the cost of agreed changes in scope.

- Together with tables 3.7.11, 3.7.12 and 3.8.2, table 4.2 and the Big Hitters 6 and 7 forms the *"Reference model success and failure factors ICT projects" (for short: SUFFI model, SUFFI = SUccess/Failure Factors in ICT projects) (see figure 10.1).* That establishes "The opinion of others about SUFFIs in Software Project Management".

- The Big Hitters are:
    - poor project management;
    - deadlines are unrealistic;
    - poor communication;
    - incomplete/weak definition requirements;
    - insufficient involvement of future users;
    - lack of senior management involvement and commitment;
    - lack of professionalism. .

- The SUFFIs that are applicable to the particular projects from the author's portfolio are:
  - table 5.2.1: Success/failure factors POTVIS project;
  - table 5.3.1: Success /failure factors Kolibrie project;
  - table 5.4.2: Success/failure factors Charging method project;
  - table 5.5.1: Success/failure factors Telephony project;
  - table 5.6.1: Success /failure factors OKAPI project;
  - table 5.7.1: Success /failure factors GIRAF project;
  - table 5.8.1: Success/failure factors AUBID project;
  - table 5.9.1: Success/failure factors VDV project;
  - table 5.10.1: Success/failure factors BIBLIOSYSTEM project;
  - table 5.11.1.1: Success/failure factors NUMIS-2000 project;
  - table 5.11.2.1: Success/failure factors SYSA project;
  - table 5.11.3.1: Success/failure factors ACCINT project;
  - table 5.11.4.1: Success/failure factors FINANCE projects.

- Although the results are based on a very limited spot check, the conclusion may be drawn that the collection Big Hitters within this collection of cases acts discriminating. Where at least four of the five Big Hitters are not applicable, the "score" is positive. Where at least four of the five Big Hitters are applicable, the "score" is negative. This picture also corresponds with the author's experiences and observations during other projects and audits. When the five Big Hitters lead to a negative score, a large number of other SUFFIs usually play a part.

- It is possible to present the SUFFIs in a SUFFI Chart that is easy to use in practice by others.

- It is possible to apply the SUFFI Chart in the "Netherlands Court of Audit" case.

10.4 FURTHER RESEARCH

For reasons of major social interest, the subject of success/failure factors will demand the necessary attention over the next few years. Both researchers as well as project managers are invited to test the SUFFI model/chart against their own research results and practical experience. This will enable further improvement of the SUFFI model/chart, serving all those involved in ICT projects well.

I propose a thesaurus of SUFFIs (similar to the ISO 9126-standard quality model).

10.5 RECOMMENDATIONS

Management, project management, project officers, (future) users can learn from others by studying the SUFFI Chart carefully. That does not take a lot of time. The seven Big Hitters are of major importance to all projects:

- poor project management;
- deadlines are unrealistic;
- poor communication;
- incomplete/weak definition requirements;
- insufficient involvement of future users;
- lack of senior management involvement and commitment;
- lack of professionalism.

Studying the SUFFI Chart leads to a collection of SUFFIs that are relevant to the project (management) in question. In addition to the Big Hitters, select a project specific top ten of SUFFIs. Reread the SUFFI Chart during the project on a regular basis and amend the collection of project SUFFIs if necessary.

Some more (author's) recommendations related to SUFFIs from the SUFFI Chart:

- prepare the project thoroughly (¬BH/01, ¬JS/RC15, ¬JS/RC16);
- be sure to have a Business Case (PN/06);
- make a risk analysis based on the Big Hitters and the top ten of SUFFIs and take countermeasures if necessary (¬JS/PUBRC05, ¬EV/07);
- refuse to start when the deadlines are unfeasible or adjust these after consultation with the client (¬BH/02);
- make realistic plans, not "desired" plans (¬JS/RC07, ¬JS/RC15);
- if necessary, adjust plans on time and substantiated. Do not allow the project to run its course (¬JS/RC15, ¬BH/01);
- do not start a project that is too sizeable or cut the project into pieces (¬BH/01, ¬JS/RC06);
- work methodically but use the method as a tool, not as a goal (JRR/02, TG/01, ¬PN/16);
- wherever possible, deliver the project results in phases (¬JS/RC06, ¬JS/PUBRC03);
- ensure an adequate way of communicating with the various parties (draw up a communication plan (¬BH/03, ¬NB/06, ¬KY/08);
- make sure that the embedding of the project results in the user organisation is arranged and approved (¬JS/PUBRC02, ¬BH/05, EX63);

- work on your own quality through continuous education (PN/10, ¬EX49);
- be convinced that every project will encounter problems that need to be resolved and view these as a challenge. Be creative with regard to solutions (¬PN/04, PN/10, ¬EX6).

Some more (author's) recommendations related to *new SUFFIs*:

- try to work with small, expert teams (AvD/01*);
- look for projects that also appeal to the project leader as regards to application. Preferably not too sizeable but certainly challenging (AvD/02*);
- a good project leader rarely has problems with his project officers and manages to motivate them well (TAH/16, AvD/03*);
- complete the project properly, it will make you feel good (AvD/04*);
- avoid wherever possible "big bang" scenarios (AvD/05*);
- make sure that the project officers are enjoying themselves (challenge) (TAH/16, AvD/06*);
- involve ICT Management in the project at an early stage (requirements from ICT Management) (AvD/07*);
- budget for having a few audits done (AvD/08*);
- as a project leader, ensure a good project administration but spend the bulk of your time by far on project officers and project aspects with regard to content (AvD/09*);
- go for quality and do things right in one single go (AvD/10*).

*) These new SUFFIs AvD/01 to AvD/10 inclusive can be added to the SUFFI Chart and the SUFFI Total Chart. The new SUFFIs AvD/01 to AvD/04 are already mentioned in section 9.2

ABBREVIATIONS

ACCINT	ACCess INTernet
ADS	Application Development System
AO	Administratieve Organisatie
APM	Algemeen Project Manager
ASZ	Automatisering Sociale Zekerheid
AUBID	AUtomatisering BIbliotheek DUT
AvD	Aart van Dijk
B&B	Beheer & Bediening
BBNP	Basis Beveiligingsniveau Nederlandse Politie
BEAdap	Baan Engin Adapter
BH	Big Hitter
BI	Beheer Infrastructuur
BIA	Bestuurlijke Informatie en Automatisering
BIBINFO	BIBliotheek INFOrmatiesysteem
BOW	Baan Open World
BR	Basic Register
BSW	Business Service Work
BUBC	Business Unit BedrijfsCommunicatie
CCA	Clinger Cohen Act
CCIP	Changes Configuration Incidents Problems
CDC	Control Data Corporation
CDD	Commissie Derde-Deskundigen
CEO	Chief Executive Officer
CFO	Chief Financial Officer
CIA	Confidentiality, Integrity, Availability
CICS/VS	Customer Information Control System/Virtual Storage
CIO	Corporate Information Officer
CIT	Computer Integrated Telephony
CJ	Capers Jones
CKR	Centraal Klanten Register
COBOL	COmmon Business Oriented Language
COSO	Committee Of Sponsoring Organisations of the Treadway Commission
CPU	Central Processing Unit
CRM	Customer Relationship Management
CSMT	CICS Master Terminal transactie
DABC	Dienst ABC
DAISY	Delfts Algemeen Informatie SYsteem
DB/DC	Data Base / Data Communication
DBH	Dienst Bouw en Huisvesting
DBMS	Data Base Management System
DINF	Dienst Informatieverzorging

DIR	Director
DMS	Document/Record Management System
DOBIS	Dortmund Bibliotheks-system
DSDM	Dynamic Systems Development Method
DTA	Develop Test Acceptance
DUT	Delft University of Technology
EAI	Enterprise Application Integration
EAP	Edp Audit Pool
EMITA	Executive Master of IT Auditing
EngD	Engineering Doctorate / Doctor of Engineering
ERP	Enterprise Resource Planning
ESS	Employee Self Service
EUR	Erasmus University Rotterdam
ExCons	External Consultant
EV	Evans
EVO	Evolutionary project management
FID	Functional Interface Document
FIT	Field Information Terminal
FP	Function Point
FU	Functions
GAK	Gemeenschappelijk Administratie Kantoor
GIRAF	General Information RetrievAl Facilities
GO	GebruikersOrganisatie
GSM	Global System for Mobile Communication
HIPO	Hierarchical Input Process Output
HOPER	Head of ICT-Operations
HOPS	Head Operations
HRM	Human Resource Management
HSBI	HoofdStuurgroep Bestuurlijke Informatievoorziening
I&I	Inventarisatie & Implementatie
IAF	Integrated Architecture Framework
IBM	International Business Machines
ICE	Integrated Computer Engineering
ICT	Information and Communication Technology
ID	Identification
IDMS	Integrated Database Management System
IMPALA	Information Management PAbx and Local Administration
IOI	Issues of Influence
Ir.	Ingenieur (Master of Science in Engineering)
ISDN	Integrated Services Digital Network
ISO	Informatie Systeem Ontwikkeling
IT	Information Technology

ITIL	IT Infrastructure Library
JO	Jan Ooninex
jp	journal publication
JR	John S. Reel
JRR	Jaap R. van Rees
JS	John Smith
KAD	Kwaliteit Administratieve Dienstverlening
KJM	Klaas-Jan Molendijk
KLPD	Korps Landelijke Politie Diensten
KPN	Koninklijke Ptt Nederland
Kolibrie	Kpn On line Interfacing met Baan, Resultaat Is Effectief
KY	K.T. Yeo
LEV	Leverancier
LM	Lorin May
M&M007	Meld- en Meetpunt 007
ME	Michael Evans
MHA	Multihouse Automatisering B.V.
MIPS	Million Instructions Per Second
MIS	Management Information System
MSc	Master of Science
MSF	Microsoft Solution Framework
MVO	Marketing, Voorlichting & Opleiding
MVS	Multiple Virtual Storage
NB	Nico Beenker
NGI	Nederlands Genootschap voor Informatica
NIST	National Institute of Standards and Technology
NOREA	Nederlandse Orde van Register EDP/IT Auditors
NWO	Nederlandse Organisatie voor Wetenschapppelijk Onderzoek
NYSE	New York Stock Exchange
OKAPI	Online KAPItaaldienst
PABX	Private Automatic Branch eXchange
PB	Productteam Basisregistraties
PC	Personal Computer
PERSIS	PERSoneels Informatie Systeem
PhD	Doctor of Philosophy
PID	Project Initiation Document
PM	Project Manager
PN	Peter Noordam
POTVIS	PrOjecT Verbetering Infrastructuur Sap
PRINCE2	PRojects IN Controlled Environments 2
PROGA	Programma A
PROGB	Programma B
PUBRC	Root cause in the UK public sector

PVB	Programma Verbetering Bedrijfsvoering
QA	Qualtiy Assurance
RAD	Rapid Application Development
RBAC	Role-Based Access Control
RC	Root cause
RC	Rekencentrum
RE	Research Engineer
RE	Registered EDP-/IT-auditor
RIP	Regeling Informatiebeveiliging Politie
RUL	Rijksuniversiteit Leiden
RUP	Rational Unified Process
SAP	Systems, Applications and Products in Data Processing,
SARA	Stichting Academisch Rekencentrum Amsterdam
SBI	Stuurgroep Bestuurlijke Informatievoorziening
SDM	System Development Methodology
SEM	Stuart E. Madnick
SGOA	Stichting Geschillen Oplossing Automatisering
SIS	Studenten Informatie Systeem
SNA	Systems Network Architecture
SOI	Spheres of Influence
SOX	Sarbanes-Oxley Act
SUFFI	SUccess/Failure Factors in ICT
SWV	SamenWerkingsVerband NUMIS-2000
SYSA	System A
TAH	Tarek Abdel-Hamid
TCS	Transaction Counting System
TG	Tom Gilb
TOGAF	The Open Group Architecture Framework
TP	TeleProcessing
TPM	Technical Project Manager
TS	Telephony Services
UA	Unit A
UB	Unit B
UK	United Kingdom
UML	Unified Modelling Language
UoA	University of Amsterdam
USA	United States
VDV	Verkeers Data Verzameling
VUP	Vax Unit of Performance
WAS	WEB Application Server
XML	eXtensible Markup Language
XP	Extreme Programming

REFERENCES

Aarts, Emile, 2006/2007, *Software Engineering*, Joint Academic Quality Research and Development (Jacquard), NWO, Den Haag

Abdel-Hamid, Tarek and Stuart E. Madnick, 1991, *Software Project Dynamics: An Integrated Approach*, Prentice Hall

Ackoff, R.L., 1971, *A concept of corporate planning*, Wiley, London

Beenker, Nico, 2004, *Studie naar succes- en faalfactoren van complexe ICT projecten*, Vrije Universiteit Amsterdam, May

Belkum, J.W. van, 1979, *Automotisering van de informatieverzorging*, Samsom Uitgeverij, Alpben a/d Rijn

Berghout, Egon, 2002, *Informatietechnologie, een bodemloze put?*, Intreerede Rijksuniversiteit Groningen, 22 oktober

Boddie, John, 1987, *Crunch Mode*, Prentice Hall/Yourdon Press

Brooks Jr., Frederick P., 1995, *The Mythical Man-Month* (20th Anniversary Edition), Addison-Wesley

Business & IT Trend Institute, 2007, *Projectmanagementenquête 2006/2007. Handvatten voor succesvolle projecten*, Vrije Universiteit Amsterdam / BISNEZ Management, Bergambacht

Dalcber, Darren and Lindsey Brodie, 2007, *Successful IT Projects*, Middlesex University Press, London

Davis, Alan, 1993, *Software Lemmingineering*, IEEE Software September, pp. 79-81,84

Dearden, John, 1966, *Myth of real-time management information*, Harvard Business Review, May/June, p. 123

DeMarco, T., 1972, *Mystical Information Systems*, Prentice Hall Inc., New York

DeMarco, Tom, 1995, *Why Does Software Cost So Much?*, Dorset House Publishing, New York

Derksen, B. and P. Noordam, 2006, *Modellen die werken, kwaliteit in bedrijf en informatievoorziening*, Boekdrukkunst uitgeverij, Gouda

Dickson, G.W. and J.K. Simmons, 1970, *The behavioural side of MIS*, Business Horizons, August, pp. 59-71

Dijk, A.J. van and M. Looijen, 1977, *Verslag van de preliminaire studie van het systeem DOBIS/LIBIS in relatie tot de Centrale Bibliotheek en het AUBID-project*, Computer Centre, Delft University of Technology, Delft, April

Dijk, Aart J. van, 1982, *Boekrecensie "Waarom falen informatiesystemen nog steeds?*, Lekturama, Voorburg

Dijk, A.J. van, P. Makkes and R.J.J. van Schie, 1983, *Definitiestudie GEMS – CEM*, Delft University of Technology, Delft, January

Dijk, A.J. van and J. Keppel, 1986, *UIVER, Globaal ontwerp van een verkiezingsinformatiesysteem bij de Universiteit van Amsterdam*, Amsterdam, September

Dijk, Aart van, Maarten Looijen and Gert Nielen, 1997, *Onderzoek NUMIS-2000 ten behoeve van de Arrondisementsrechtbank te Utrecht*, Avédé-Info, Zoetermeer, February

Dijk, Aart J. van, 2001, *Complexiteit, pragmatiek en beheer*, in: Complexiteit van Beheer, Liber Amicorum voor prof.dr.ir. M. Looijen, Delftse Universitaire Pers, Delft, May

Dijk, Aart J. van, 2004-1, *Onderzoek/audit naar de status quo van het SAP-project SYSA*, GOVERN, Rotterdam, March

Dijk, Aart J. van, 2004-2, *Onderzoek/audit naar de gang van zaken van het project ACCINT, Uitrol Browse-functionaliteit/e-mail-functionaliteit*, PUBLIC, ICT-Operations, Amsterdam, July

Dijk, Aart J. van, 2006, *FINANCE en SOX 2006, Werkwijze, resultaten, evaluatie en advies*, FINANCE, ICT Support, Groningen, December

Evans, Michael W., Alex M. Abela and Thomas Beltz, 2002, *Seven Characteristics of Dysfunctional Software Projects*, CrossTalk, April, pp. 16-20

Field, T., 1997, *When BAD Things Happen to GOOD Projects*, CIO, 15 October, pp. 55-62

Flowers, S., 1996, *Software failure: management failure*, John Wiley, Chichester, UK

Frielink, A.B., 1976, *Voorbereiding van computertoepassingen*, Academic Service, Den Haag

Gilb, Tom, 1988, *Principles of Software Engineering Management*, Addison-Wesley Publishing Company

Gilb, Tom and Lindsey Brodie, 2005, *Competitive Engineering*, Elsevier, Oxford

Glass, Robert, 1998, *Software Runaways – lessons learned from massive software project failures*, Prentice Hall

Hammink, R.J., 1979, *Administratieve Automatisering in Nederland 1976-1981*, Studiecentrum NOVI, Amsterdam

Hendriks, Brigit, Mark van Onna and Günther Schraven, 1997, *De kleine Prince 2 – Projectmanagement methodiek voor kleine en middelgrote projecten*, Pink Elephant, Zoetermeer

House of Commons, 1999, *First Report: Improving the delivery of government IT projects*, Session 1999-2000, Committee of Public Accounts

Humphrey, Watts, 1998, *Three Dimensions of Process Improvement: Part I: Process Improvement*, CrossTalk, February, pp. 14-17

Institute for Defence Analysis, 1994, *Technical Risk Indicators for Embedded Software Development*, Institute for Defence Analysis, Paper P-3027, October

Jones, Capers, 1994, *Assessment and Control of Software Risks*, Yourdon Press, Upper Saddle River

Jones, Capers, 1996-1, *Large Software System Failures and Successes*, American Programmer, April, pp. 3-9

Jones, Capers, 1996-2, *Patterns of Software Systems Failure and Success*, International Thomson Press

Klooster, A.J. van 't, 1968, *Administratieve Organisatie, organiseren en bestuurlijke informatieverzorging*, Samsom, Alphen a/d Rijn

Klooster, A.J. van 't and J.A.M. Oonincx, 1978, *Leerboek Informatiesystemen*, Samsom, Alphen a/d Rijn

KPMG, 2004, *Programmamanagement leidt tot betere projectresultaten*, KPMG, Utrecht

Kuypers, G., 1982, *ABC van een onderzoeksopzet*, Coutinho, Muiderberg

Louws, A., 2008, *Projectmanager Agile moet vooral coachen*, Computable, No. 43, 24 October

Lucas jr, H.C., 1975, *Why Information Systems Fail*, Columbia University Press, New York

May, Lorin J., 1998, *Major Causes of Software Project Failures*, CrossTalk, July, pp. 9-12

Mc Farlan, F. Warren, 1971, *Problems in planning the information system*, Harvard Business Review, March/April, p. 77

Mc Kinsey, 1968, *Unlocking the computer's profit potential*, Mc Kinsey & Company Inc, New York

Molendijk, Klaas-Jan and Stef Oud, 2007, *Projectmanagement is risicomanagement*, Automatisering Gids, No. 47, 23 November, pp. 18-19

Neering, Rob, Frank Harmsen, Ronald Batenburg, Eric Bunk and Sjaak Brinkkemper, 2006, *De methode doet het nog steeds*, Informatie, March, pp. 18-26

Nielen, G.C. and S.F. Nielen, 1996, *Het georganiseerde informatiesysteem*, Handboek Informatica, Samsom, Alphen a/d Rijn, August

Noordam, Peter, Norman Martijnse and Berry Derksen, 2007, *ICT-projectmanagement op weg naar volwassenheid*, Informatie, March, pp. 20-26

Oonincx, J.A.M., 1982, *Waarom falen informatiesystemen nog steeds?*, Samsom, Alphen a/d Rijn

Rathenau, Rapport van de Adviesgroep, 1980, *Maatschappelijke gevolgen van de Micro-electronica*, Staatsuitgeverij, Den Haag

Reel, John S., 1999, *Critical Success Factors In Software Projects*, IEEE Software, Vol. 16, No. 3, May/June, pp. 18-23

Rees, Jaap R. van, 1982, *De methode doet het niet*, Informatie, February, pp. 81-93

Rijsenbrij, D.B.B. and H.J. Kouwenhoven, 1996, *Projectmanagement-audit*, Academic Service, Utrecht

Smith, John, 2001, *Troubled IT projects: prevention and turnaround*, IEE, Herts, UK

Smith, John, 2002, *The 40 root causes of troubled IT projects*, Computing & Control Engineering Journal, Vol. 13, No. 3, June, pp. 109-112

Software Program Managers Network, 2000, *16 Critical Software Practices For Implementing Performance-Based Management*, Ver. 3.0, Integrated Computer Engineering, Inc., Arlington, Va., 2 August

The Standish Group International Inc., 1995, *Standard Group CHAOS Report*, Boston

The Standish Group International Inc., 1998, *Standard Group CHAOS Report*, Boston

The Standish Group International Inc., 2003, *Standard Group CHAOS Report*, Boston

Verhoef, Chr., 2007, *Goed IT-bestuur begint bij minimale eisen*, Digitaal bestuur, Vol. 3, No. 9, October, pp. 48-51

Verschuur, J., 2003, *ICT Barometer*, jaargang 3, Ernst & Young, Amsterdam

Vink, W., 2002, *De SGOA in vogelvlucht. Een verslag over 200 zaken uit de jaren 1991 t/m 2001*, Rotterdam

Volkskrant, 1997, *Multihouse ten onder na grote schadeclaim*, Volkskrant 19 April, Amsterdam

Vrede, T. de, 2001, *Wonderen blijven nog uit. Veel CRM-projecten mislukken*, AG, No. 26, 28 June

Vrede, T. de, 2002, *CRM stelt tot nu tae teleur*, AG, No. 49, 5 December

Westerveld, Wijnand, 1997, *Multihouse verslikt zich in molensteen Numis*, Automatisering gids, No. 17, 25 april

Wijsman, T., F.H. Band, R.S. de Boer, A. Colon, N. Donker, M.E.M. Kerkvliet and E. Leydesdorff, 2007, *Lessen uit ICT-projecten bij de overheid*, Algemene Rekenkamer, Den Haag, November

Yeo, K.T., 2002, *Critical failure factors in information system projects*, International Journal of Project Management 20, pp. 241-246

Yourdon, Edward, 2004, *Death March*, second edition, Yourdon Press

Zutpben, L.C. van, 1973, *Interne cantrole en beveiliging van geautomatiseerde informatiesystemen*, Vermande Zonen Uitgevers, IJmuiden

REFERENCES OF "SOFTWARE PROJECT DYNAMICS – AN INTEGRATED APPROACH"

9. Abdel-Hamid, T.K. and Madnick, S.E., *A Model of Software Project Management Dynamics*, The 6th International Computer Software and Applications Conference (COMPSAC), November 1982

19. Alberts, D.S., *The Economics of Saftware Quality Assurance*, National Computer Conference, 1976

22. Anthony, R.N. and Dearden, J., *Management Control Systems*, Chicago, IL: Richard D. Irwin, Inc., 1980

23. Aron, J.D., *Estimation Reasons for Large Programming Systems*, Software Engineering: Concepts and Techniques. Edited by J.M. Buxton, P. Naur and B. Randell. Litton Educational Publishing, Inc., 1976

24. Arseven, S.M., *A System to Monitor and Contral the Development and Documentation of a Computer programming Project*. Unpublished PhD. Dissertation, Texas A&M University, 1975

28. Baber, R.L., *Software Reflected*. New York, NY: North Holland Publisbing Company, 1982

33. Barbacci, M.R. Habermann, A.N. and Shaw, M., *The Software Engineering Institute: Bridging Practice and Potential*. IEEE Software, November 1985, pp 4-21

36. Basili, V.R., *Improving Methodology and Productivity Through Practical Measurement*. A Lecture at the Wang Institute of Graduates Studies, Lowell, M.A., November 1982

38, 41, 42, 43, 44, 46, 47, 49, 54, 55, 57, 61, 62,
63, 65, 68, 69, 71, 72, 74, 75, 76, 77, 78, 80, 81,
84, 85, 86, 87, 88, 89, 90, 92, 94, 96, 99, 101, 102,
103, 105, 109, 110, 111, 113, 116, 117, 119, 123,
125, 130, 131, 136, 137, 139, 144, 145, 146, 158,
161, 165, 166, 167, 169, 172, 173, 177, 178, 180,
181, 182, 184, 188, 189, 190, 200, 207, 208, 210,
213, 215, 218, 223, 224, 225, 227, 228, 230, 231,
234, 236, 238, 240, 241, 242, 243, 244, 246, 249,
252, 256, 259, 260, 261, 265, 270, 272, 273, 274, 275*)

*) Reference number in Bibliography of

" Tarek Abdel-Hamid and Stuart E. Madnick,
Software Project Dynamics: An Integrated Approach,
Englewood Cliffs, NJ: Prentice Hall, 1991"

APPENDICES

APPENDIX 1

THE PUBLICATION OF JAN OONINCX [1982]: WHY ARE INFORMATION SYSTEMS STILL FAILING?

Introduction

Professor Oonincx: "In this publication it is discussed why up to date (1982) so few successfully computerised information systems were realised [Oonincx 1982, Lucas 1975]. Through pointing out a number of responsible causes (the errors and shortcomings from the past), we do hope to contribute to different (adjusted) ways of approaching the development of information systems. In this book, the lack of a theory and methodology for developing information systems was selected as a starting point."
In this appendix, the 17 chapters will be discussed in brief.

The lack of a methodology for developing information systems

- the traditional methods of system development are not suitable for setting up modern and voluminous information systems [Van Zutphen 1973];
- the major errors are:
    - the information systems were set up too ambitiously [Van 't Klooster & Oonincx 1978];
    - the information systems were set up too separately (islands of automation). All this was the result of a lack of an overall perception;
    - the lack of proper planning of information systems;
    - the relation between the information sub systems and the data collections were insufficiently researched;
- using an incorrect design philosophy and design methodology leads to shortcomings and/or inaccuracies in the system set-up, which may also have more or less serious consequences for the reliability of the entire information system;
- one should pursue the set-up of a coherent arrangement of information sub systems.

Insufficient insight into decision-making processes and information requirements

- when making decisions, the manager is to an important degree dependent on the availability and accessibility of relevant information;
- it has turned out that managers are not at all or hardly able to indicate how they proceed when making decisions;

- Ackoff [1971] denies that managers are able to state exactly what information they need for taking effective decisions;
- the information system should never be perceived as an objective in itself;
- the information system should be both registering as well as steering;
- nowadays, the technical developments in computerisation have progressed such that allocation of local power of decision may be coupled with central data acquisition and data processing.

The feasibility of integral information systems

- it is obvious that those involved in designing and introducing information systems are interested in the problems and difficulties that may hinder the realisation of the objectives;
- after 1970, increasingly more books and articles appeared in which the authors adopt a very sceptical attitude with regard to the applicability's of information systems that cover the entire business [DeMarco 1972, Mc Kinsey 1968, Dearden 1966];
- it is hardly possible to develop one single comprehensive information system. Therefore, it is obvious to divide the entire information system into a number of information sub systems that can be realised.

Insufficient involvement and cooperation of users in setting up information systems

- whether and to what extent the new information system will become a success depends on the question whether the users have been given sufficient opportunity to specify their information needs and of the manner in which the users are involved in development of the information system;
- attention has to be drawn to the fact that the top management is sometimes insufficiently involved in the development of information systems. If the top management adopts a passive attitude it will be very difficult, if not impossible, to get an information system suitable for that particular business off the ground;
- the new information system will have to be the result of joint effort and collaboration of the users, system analysts, computer specialists etc. It needs to be avoided that the development process is viewed as a technical affair;
- it is known that changes, and the introduction of a computerised information system is a considerable change and may go accompanied with feelings of discomfort for all those involved [Dickson & Simmons 1970].

Insufficient and/or inefficient use of the possibilities as offered by the computer in the information system

- notwithstanding the general applicability and the universal nature of computers, in many businesses the uses of the computer from its introduction at the start of the Fifties remained limited to applications in the administrative sphere for a number of consecutive years [Frielink 1976];
- the Eerste Landelijke Automatiseringsenquête (First National Computerisation survey) in 1979 [Hammink 1979] shows that a high level of automation has been achieved for traditional areas of application. However, with regard to more advanced applications the level of automation is considerably less (between approximately 30% and 60%, dependent on the size of the organisation);
- indication of the effectiveness of computerised information systems is complex and therefore very difficult and will as a rule include subjective elements [Mc Farlan & Warren 1971];
- research [Mc Kinsey 1968] has shown that businesses with good plans for all computer applications arrived at more effective information systems than businesses that approached these activities less systematically;
- when analysing the information; tracking of trends, discovering coherences and such like, the possibilities of the computer were discovered at a much later stage.

Not using the new information system

- if a once designed and introduced information system is not actually used, one may already speak of a failure;
- only when the users fully accept the new information system, the advantages as expected can be realised;
- an effect of not or just partial use of the new information system is that the supplied information does not suit the purposes for which it is demanded.

Collection and storage of too much and unsuitable data

- the output of the information system should be determined by the information needs;
- only that data needs to be collected that by means of the information system contributes to satisfying the information requirements;
- the information system should not be viewed as a means to record all sorts of data that might be useful for decisions to be made in future. This results in too little attention to the selective collection of the required data.

Ineffectiveness and limited applications of traditional systems for data storage

- the effect of island automation of individual applications and information sub systems was that every application acquired its own data collections. The consequences were that the very same data was stored in several locations in the files of the individual information systems;
- in practice, the problems with regard to conversions are substantial (malfunctions, time-consuming, costly);
- in principle, one should pursue once-only storage of data as referred to by the programmes of several information sub systems.

Set-up of information systems too static

- at construction of information systems, too little thought was given to the consequences of changing circumstances from within the business as well as from the outside. It becomes very difficult to modify designs at a later stage once these are determined;
- in a number of cases it has happened that businesses through lack of financial means, started with a smaller computer configuration than would have been sensible in view of the growth of the business. Once the growth was realised, the computer proved to be too small;
- in order to be able to react decisively to changes, the information system will need to be flexible. At construction of the information system one has to be prepared for future changes [Van 't Klooster 1968]. The flexibility indicates the degree to which the information system is capable of reacting to cbanges;
- flexibility comes into being when the data is stored separate from the application for which it is needed;
- the planned overcapacity at procurement should remain limited to that extra space, which is expected to be required for enabling development over the first few years.

Insufficient suitable human resources for development and implementation of information systems

- currently, there is still a very large need for automation experts and specialists in the field of information systems with higher vocational training or a university background;
- in 1982, there is still a vast shortage of well-trained (managerial) information scientists in the Netherlands;
- the Information science training has to be promoted strongly in order to comply with the quantitative and qualitative needs for information scientists [Rathenau 1980].

The influence of changes to the information system on the organisation

- every effort has to be made for integrating the information and technological developments with the changes in the organisational structure;
- the development and introduction of a new information system in an organisation means a change process. In this change, one comes up against the behaviour of humans within the organisation. Evidently, people do often have problems with such change processes;
- the possible negative effects of the resistance against change processes have to be avoided wherever possible by means of good advice and support.

Too little attention to the informal aspects in relation to the information system

- no organisation is able to function without the existence of informal data flow and informal relationships;
- in order to obtain the best possible and complete idea of the information needs, one should also attempt to include the informal aspects of the information system in the analysis of the information requirements;
- from a cost point of view, one should also keep an eye on the informal organisation and the informal data flow. Therefore, one should look the other way with regard to these but also make sure that these are kept within certain boundaries.

The development process takes too long and the costs are too high

- control over the development process is an essential condition for the success of the information system;
- a few problems connected to the long duration of the building process are:
    - that the demands that are made on the information system can change in the intervening period;
    - the staff turnover in the automation sector resulting in repeatedly having to provide staff with induction and documentation problems arising etc;
    - the general involvement of the users as time goes by and introduction fails to materialise;
    - the development costs that extensively exceed the original budgets;
- the entire process has to be divided into phases.

Computerisation based on status considerations

- computerisations based on status considerations is fundamentally wrong;
- however, once it is installed and functioning properly the information system may provide the owner with a certain prestige.

Too little attention to the aspect of profitability and the lack of measuring instruments or determining the benefits of the Information system

- a new or modified information system should not just be technically feasible and practically applicable but it should also be desired from an economical point of view;
- the economical feasibility can only be determined through a cost benefit analysis. With regard to information and information systems, the item cost benefit analysis is still in its infancy;
- it is necessary to work out the objectives per information system. Within that framework, it will also be necessary to lay down "standards". That way it is also possible to check afterwards whether the objectives (and the benefits) have been achieved;
- in the elaboration, it will have to be determined in each phase whether one is able to continue;
- at first when it is put into use and later on periodically, the entire information system will have to be investigated and evaluated on efficiency and effectiveness.

Unsuitable equipment and software

- whilst up to the seventies it was conceivable that the technical possibilities caused limitations and often were the "bottle-neck" in the uses of computers in information systems, these days it is no longer possible to state that the available equipment needs to cause any major problems;
- any shortcomings can be explained sooner from the application programs. Because these cannot be any better than the system design and the methods for system building allow;
- especially important or the program design and the program development are techniques such as structured programming, modular programming and interactive programming;
- the organisation has to determine first whether suitable ready-made programs are available for certain applications, before having their own programs designed for that purpose.

Insufficient attention to reliability and controllability

- in the past when building information systems, too little importance was attached to the reliability and controllability aspect;
- if the computer processing does not lead to the intended results, this is almost always caused by human failings with regard to the system set-up, programming, data input, operation of the computer or by the lack of adequate control measures;
- a reliable system of internal control and physical safety measures is indispensable for realising reliable information systems [Van Belkum 1979].

APPENDIX 2

THE PUBLICATION OF TAREK ABDEL-HAMID AND STUART E. MADNICK [1991]: SOFTWARE PROJECTS DYNAMICS – An INTEGRATED APPROACH

1 Introduction

For each chapter the main subjects (and accompanying sentences) of "*the procedures in Tarek Abdel-Hamid's work on Software Project Management*" [Abdel-Hamid & Madnick 1991] have been mapped point by point in this appendix. Based on this information, SUFFIs are derived.

2 The chapters of "Software Project Dynamics – An Integrated Approach"

2.1 Preface

- The development of software systems has been plagued by cost overruns, late deliveries, and users' dissatisfaction.

- The objective of this book is to enhance systematically the understanding of and gain insight into the general process by which software development is managed.

- Abdel-Hamid (TAH) and Madnick (SEM) developed an integrative model of software development project management. The model was developed on the basis of an extensive review of tbe literature supplemented by 27 focused field interviews of software project managers in five organisations.

- This model divides the software development and management activities into four areas: (1) human resource management, (2) software production, (3) controlling, and (4) planning. Over 100 individual but interdependent phenomena were identified and represented using the system dynamics modelling notation.

- A case study in a sixth organisation was conducted to test the model.

- The model was used as an experimentation vehicle to study or predict the dynamic implications of an array of managerial policies and procedures. Four areas were studied: (1) scheduling, (2) control, (3) quality assurance, and (4) staffing.

2.2 Chapter 1: Introduction

- The software industry has been marked by cost overruns, late deliveries, poor reliability, and users' dissatisfaction {103,182,228,249} (p3).

  {…} = reference number in the list of references included in the book
  (p...) = page in the book

- Software is often on the critical path in overall system development (p3).

- Software projects are sometimes considered successful when the overruns are held to thirty percent or when the user only junks a quarter of the result (p4).

- Mitch Kapor (Lotus Development Corporation) believes that software design must be improved and the development process better understood {63} (p4).

- The new discipline is called "software engineering". It encompasses both the technical aspects of software development (e.g., design, testing, validation) as well as the managerial ones {33,47,246} (p4).

- There are more opportunities for improving software productivity and quality in the area of management than anywhere else {46} (p5).

- The basic problem is management itself {112}. Poor management can increase software costs more rapidly than any other factor {261} (p5).

- There is still lack in the fundamental understanding of the software development process {87,96,173} and without such an understanding the likelihood of any significant gains in the management of software development front is questionable {36,130,161} (p5).

- The objective of this book is to develop and test an integrative view of software development project management in order to enhance the understanding of, provide insight into, and make predictions about the general process by which software development is managed (p6).

- There are hundreds of variables that affect software development. Furthermore, these variables are not independent; many of them are related to one another {119,189} (p7).

- The two key features of this model that distinguish it from most others in the software engineering are that: (1) it is integrative, and (2) it is a system dynamics model (p7).

- The model is integrative in the sense that it integrates the multiple functions of the software development process, including the management-type functions (e.g., planning, controlling, and staffing) as well as the production-type functions that constitute the software development life cycle (e.g., designing, coding, reviewing, and testing) (p7).

- The behaviour of an individual subsystem in isolation may be different from its behaviour when it interacts with other subsystems {68} (p8).

- The system dynamics philosophy is based on several premises:
    - the behaviour (or time history) of an organisational entity is principally caused by its structure;
    - managerial decision-making takes place in a framework that belongs to the general class known as information-feedback systems;
    - intuitive judgment is unreliable about how these systems will change with time, even when we have good knowledge of the individual parts of the system;
    - thorough model experimentation is possible to fill the gap where judgment and knowledge are weakest – by showing the way in which the known separate system parts can interact to produce unexpected and troublesome over-all system results (p9).

- Based on the above philosophical beliefs, two principal foundations for operationalising the system dynamics technique were established. These are:
    - the use of information-feedback systems to model and understand system structure;
    - the use of computer simulation to understand system behaviour (p9).

2.3 Chapter 2: Key components of software development

- The *managerial* aspects of software development have attracted much less attention from the research community {41,249,275}. Cooper provides an insightful explanation: Perhaps this is so because computer scientists believe that management per se is not their business, and the management professionals assume that it is the computer scientists' responsibility {71} (p15).

- Software development and its related project management activities are often based on the simple "mental picture" captured by a single-loop model {225} (p16).

- Brooks'Law: adding more people to a late software project makes it later {57} (p17).

- People under time pressure don't work better, they just work faster. ... In the struggle to deliver any software at all, the first casualty has been consideration of the quality of the software delivered {81} (p17).

- In this book TAH&SEM build upon and extend what has been learned about the micro components in order to construct *a holistic model* of the software development process. It integrates the multiple functions of software development, including both managerial functions (e.g., planning, controlling, and staffing) as well as software production activities (e.g., designing, coding, reviewing, and testing) (p19).

- A unique feature of the model is its use of the feedback principles of system dynamics to structure and clarify the complex web of dynamically interacting variables (p19).

- Looking within the model's boundary (e.g., at the actions of the software development team) for the causes and cures of problematic behaviour rather than outside it (e.g., the actions of the users) is a characteristic of the system dynamics approach. Richardson and Pugh called it the "Endogenous Point of View" (p20)

- TAH&SEM's primary focus is that of medium-sized projects. Jones defined medium-sized software projects as follows: "... (they) range between 16K and 64K lines in size, (and in which) development teams or departments are the norm ..." {137} (p21).

2.4 Chapter 3: Review of relevant literature

- Two factors significantly influence the initial estimate of the job size: (1) the firm's previous experience; and (2) the general over-all tendency to underestimate the job size {223} (p31).

- System complexity grows as the square of the number of system elements; therefore, experience with a small system cannot account for all the things that will have to be done in a large system. Neither will the Experience Method apply to systems of totally different content {23} (p31).

- The software undersizing problem is the most critical road block to accurate software cost estimation ... there are no magic formulas that can be used to overcome the soflware undersizing problem {49} (p31).

- It is difficult to measure performance in programming ... (And) it is difficult to evaluate the status of intermediate work such as undebugged programs or design specifications and their potential value to the completed project {181} (p32).

- In a multi-project environment, competition for company resources becomes a significant dimension (p33).

- The longer an error goes undetected, the more extensive the necessary rework and the greater the cost. Changing design specifications after development begins also results in rework (p33).

- Thayer: "By far, the two dominant (problematic) activities are planning and controlling, which together (accounted) for 80% of the issues" {246} (p34).

- Thayer noted with interest, that "there is some disagreement between the general processing community and the project managers and developers. ... The fact that these two groups do not, in general, agree on the major issues is in itself a fundamental problem of project management" (p38).

- There are no well defined software management techniques to guarantee a successful software delivery (p38).

- Riehl developed a "planning and control framework to assist in the management of computer-based information systems development in large organisations". His model, termed the "Composite-Working Model", consisted 25 "principles" and 50 "issues" {218} (p40).

- McFarlan focused on the *differences* among software development projects. He identified three "important" dimensions for characterising software development projects:
  1. the degree of predetermined structure inherent in the project;
  2. the degree of company-relative computer technology implicit in the project;
  3. project-size, measured as man-years of effort or manpower dollars of expenditures (p40).

  McFarlan classified projects in one of eight different categories {167} (p42).

- The steps or phases in the software development life cycle are described differently by different authors, but the differences are primarily in amount of detail and number of categorisations {80,116}. To put the overall process in perspective, the percentages of resources consumed in each phase must be evaluated. Numerous authors have reported on resource consumption in each phase (p44/46).

- Inadequate planning is the primary reason for loss of control on many computer programming projects {208} (p46).

- Software estimation historically has been and continues to be a major difficulty in managing software development {85,86,130,180,208,272,274,275} (p47).

- Myers has identified several "traps" in the experience method (i.e., basing estimates on actual costs of similar past projects), namely:
  1. the relationship between cost and system size is *not linear*. In fact, cost increases approximately exponentially as size increases. Therefore, the experience method should only be applied when the sizes of the current project and past projects are equivalent;
  2. products with similar names are normally dissimilar. For instance, chances are slim that two products titled "Payroll System" have the same development costs;
  3. frequent budget manipulations by management to avoid overruns make historical cost data questionable. For example, the movement of cost from an over-budget account to an under-budget account disguises the real costs and makes future use of this data dangerous {188} (p48).

- The serious student of estimating must first be willing to probe deeply into the fascinating and complex system development process, to uncover the phases and functions of the process. Only then meaningful quantitative research and scientific analysis of resource requirements can be done {207} (p49).

- A major challenge to managers is to motivate employees (p49). A mechanism for motivation, which is attracting interest in the software engineering field, is "goal setting" {49} (p50).

- Brooks suggests that human communication in a software development project is the most significant cause of overhead {57} (p51).

- While in most production environments, control is a *standard* business practice {181}, in the production of software, control is a "perilous activity" {24,46,102,110,111,158,177,178,208,246} (p52).

- The manifestation of poor software project control has more than one form. For example:
    1. the "90% Syndrome" {28,49,81,88};
    2. the production of inadequate software, e.g., which doesn't meet user requirements {117,244};
    3. systems that are inordinately expensive {172,270}, e.g., because of unconstrained gold-plating {49,145,215,270};
    4. lack of historical software cost data bases {49,246} (p52).

- People-type factors:
    1. the "software wizard syndrome" {44}. Management sometimes abdicates its responsibility to a highly trusted software specialist, whose pronouncements are *ex cathedra*;
    2. inaccurate reporting {44,111,139};
    3. optimism {74,123,136,172,200,238}. "All programmers are optimists," Brooks remarked. They always assume that " 'This time it will surely run' or 'I just found the last bug' " {57} (p53).

- Lehman's survey of software development projects in the aerospace industry were surprising: ... 17% of the projects had no project control mechanism. And more surprisingly yet, that group fared better than average relative to on-time delivery. ... {158} (p54).

2.5 Chapter 4: Sources of information

- To build the model of software project management, TAH&SEM took three information-gathering steps:
    1. a series of interviews were conducted with software development project managers in several organisations;
    2. an extensive review of the literature was conducted;
    3. another series of interviews was conducted to refine the model (p55).

- The model is exposed to criticism, revised, exposed again and so on in an iterative process that continues as it proves to be useful. Just as the model is improved as a result of successive exposures to critics a successively better understanding of the problem is achieved by the people who participated in the process {224} (p55).

- Interview topics:
    - *Environment* (project types, sizes, hardware environment, organisational structure);
    - *Software Production* (software tools, standards, error rates, QA policy);

- *Planning* (estimating, effort distribution);
- *Control* (cootrol tools, milestones, reporting frequency);
- *Human Resources* (hiring/firing policies, training, turnover, overtime policy) (p56).

2.6 Chapter 5: Human Resource Management

- A project's total work force is assumed to consist of two work force levels, namely "Newly Hired Work Force" and "Experienced Work Force" (p63).

- Newly hired project members pass through an orientation during which they are less than fully productive {75,243} (p63).

- The training of newcomers, both technical and social, is usually carried out by the "old –timers". This is costly because "while the old-timer is helping the new employee learn the job, his own productivity on his other work is reduced {55,74,77} (p65).

- Project orientation can still be a significant drag on productivity, especially when a project lacks adequate documentation {62} (p64).

- A project that is behind schedule often suffers also from sparse and outdated documentation {113} (p64).

- The "Newly Hired Work Force" is, on average, less productive than the "Experienced Work Force" (p65). (Average Assimilation Delay" is set at 80 working days).

- Toward the end of the project there is likely to be reluctance to bring in new people, because it would take too much time to acquaint new people with the mechanics of the project, integrate them into the project team, and train them in the necessary technical areas (p67).

2.7 Chapter 6: Software production

In the model there are four primary activities in the Software Production Subsystem: development, quality assurance, rework and system testing (p69).

This subsystem is too complex to explain as one piece. TAH&SEM break it into four sectors:
1. Manpower Allocation
2. Software Development
3. Quality Assurance & Rework
4. System Testing.

- Modules and changes were initially inspected in depth but with less severity as work pressure increased and greater risks were taken to meet delivery schedules {117}. Walk throughs and inspections are usually the greatest casualties. Under schedule pressures, they are not only relaxed but often suspended altogether {94} (p71).

- A mathematical model should be based on the best information that is readily available, but the design of a model should not be postponed until all pertinent parameters have been accurately measured. That day will never come. Values should be estimated where necessary {99} (p73).

- As schedule pressure mounts, quality assurance activities are relaxed; i.e. cuts are made into the *planned* QA effort. QA activities are *not* eliminated completely (p74).

2.8 Chapter 7: Software development

- Problems of communication and motivation are responsible for inadequacies in process and for consequent losses in productivity {240} (p79).

- Three identified "resource-type" variables – the availability of programming tools, the availability of programming practices, and programmer experience – and two "task-type" variables – the programming language and the quality of external documentation – as having significant influence on productivity {230} (p80).

- Most of the factors that affect the potential productivity of a software development project, listed in the literature are constants, at least two are not: work force experience level and increases in project familiarity due to learning-curve effects {76,234,261}. The learning-curve is the rate of improvement. Reflecting on his experience at IBM, Aron estimates that the total improvement for a medium-sized project (e.g., 12-24 months long) would be a 25% improvement in productivity {23} (p83).

- The project goals and schedules can play a significant motivational role throughout the life of a software development project (p85).

- When a project is perceived to be behind schedule, people tend to work harder to bring it back on schedule. They do that by compressing their slack time and/or working over-time, thus allocating more man-hours to the project. There is a threshold beyond which employees would not be willing to work at an "above-normal" rate {169} (p86).

- A compressed slack time reduces their tolerance level for continued hard work since continued hard work means a continued "deprivation" of their slack time (p89).

- When project members perceive some "excesses" in the schedule (the case of negative schedule pressure), some excesses will be "absorbed" by the workers as "under-work" before downward adjustments are made in the project's schedule {49,131}. As with positive schedule pressures, there are limits on how "fat" employees are willing or allowed to absorb (p91).

- It is widely held that communication overhead increases in proportion to n^2, where n is the size of the team {57,180,231,236,273} (p93).

2.9 Chapter 8: Quality Assurance and Rework

- Software quality assurance includes two distinct and complementary methodologies. The first is designing a coherent, complete unambiguous, and no conflicting set of requirements. The second is review and testing of the product {84} (p95).

- TAH&SEM will assume that software design begins at the "successful completion" of a software requirement's review and that there would be no subsequent changes or modifications in the system's requirements (p95).

- Design errors in the early design phase are generated at a higher rate than are coding errors {165} (p96).

- Factors that affect the "Error Generation rate" in a software project include organisational factors (e.g., the use of structured techniques {19}, the quality of the staff {42}) and project factors {236} (e.g., complexity, size of system, language) (p96). A set of factors that play a dynamic role during software development: the workforce mix and schedule pressures (p99).

- The number of errors is defined as a function of KDSI (thousand delivered source instructions). The error rates range in value from 25 errors/KSDI (early in the project) to 12.5 errors/KDSI (at the end of the project) (p99).

- Newly hired employees are not only less productive on average but also more error-prone than their experienced counterparts {92,189} (p100).

- People under time pressure don't work better, they work just faster ... {181,213,215} (p100).

- Schedule pressures often result in the "overlapping of activities that would have been accomplished better sequentially", and overlapping can significantly increase the chance of errors {252} (p101).

- Since the objective of the QA activity is to detect errors, it becomes almost impossible to tell whether *all* those errors were in fact detected (p102).

- At the organisational level, there are seldom any rewards that promote quality or quality-related activities {72} (p103).

- The distribution of errors is pyramidal, with the majority of errors requiring a few hours to detect, a few errors requiring approximately a day to detect, and still fewer errors requiring more than a day to detect {38} (p105).

- On average a design error is approximately 1.5 times more costly to correct than a coding error {19,54,189} (p107).

- As detected errors are reworked, some fraction of the corrections will be bad fixes. Unfortunately, there are no published data on how large that fraction is (p108).

2.10 Chapter 9: System testing

- Errors that QA fails to detect while the software is being designed and coded and bad fixes resulting from faulty rework remain undetected until the system testing phase. TAH&SEM will assume that *all* such errors will get detected and corrected at the system testing phase. Thus, even though in practice some errors often remain in a software product after system testing is completed, those errors will be excluded from TAH&SEM's formulation (p109).

- Undetected errors consist of errors that escape detection of QA and bad fixes resulting from faulty rework (p109).

- Detecting and correcting a design error during design phase (i.e., through QA) is one-tenth the effort ("cost-to-fix escalation") that would be needed to detect and correct it later during the system testing phase because of an additional inventory of specifications, code, user and maintenance manuals, and so on would also have to be corrected {166} (p109).

- Although *static* estimates on cost-to-fix escalations at different points in the software life cycle are available, no data are available in the literature to describe the *dynamics* of the "error-reproduction" processes. That is, TAH&SEM know that an undetected design error reproduces enough errors in code, documentation, and so on, to become four to ten times more expensive to fix at the system testing phase, but TAH&SEM do not have data that explain exactly how and when these reproduction processes occur (p111).

- TAH&SEM assume that undetected errors will become either "Active errors" (errors that produces more errors) or "Passive errors". Because design specs are the blueprints of the system's code, any errors in design will get translated into coding errors (p111).

- The earlier the undetected error is, the more "generations" of errors it will produce, and thus the more costly it will end up being (p112).

- As the error density increases, the distribution of errors among the system's modules generally also increase; errors become less localised. They also become more expensive to detect and correct (p113).

- The objective of System Testing is to verify "that all elements (of the system) mesh properly and that overall system function and performance are achieved" {210} (p114).

- The *actual* testing effort needed per task is a function not only of testing overhead and error density, but also of how efficiently people work (p116).

2.11 Chapter 10: Controlling

- Any control function has at least three elements {22}:
  1. Measurement – detection of what is happening in the activity being controlled.
  2. Evaluation – assessment of its significance, usually by comparing information on what is *actually happening* with some standard or expectation of what *should be happening.*
  3. Communication – report of what has been measured and assessed, so that behaviour can be altered if the need for doing so is indicated (p117).

- Progress in a software project is measured by the number of resources consumed, tasks completed, or both. (remark: "or" has to be "and", AvD) (p117).

- It is difficult to measure performance in programming. It is difficult to evaluate the status of intermediate work such as undebugged programs or design specification and their potential value to the complete project {181} (p119).

- Progress, especially in the earlier phases of software development, is measured by the rate of expenditure of resources rather than by some count of accomplishments {81} (p119).

- Only when the program is almost finished or when the allocated time budget is almost used up the programmer will be able to recognise that the calculated figure is wrong {28} (p119).

- As the project advances towards its final stages accomplishments become more visible, project members become increasingly more able to perceive how productive the work force has actually been (p120).

- In the early phases of software development, progress tends to be measured by the rate at which resources are expended (p120).

- People's assumptions about their productivity change as the project develops (p121).

- As system testing gets underway, people's perceptions of their productivity become a function of how productive the testing activity *actually* is as opposed to how productive it was *planned* to be (p123).

- After an assessment is made of man-day shortages or excesses, behaviour on the project can then be altered. Adjustments are then translated into adjustments to the schedule or adjustments to the work force level or both (p123).

- As a software project develops, project members often realise that they have under-estimated the number of tasks (e.g., modules) that constitutes the software system being developed {61} (p124).

- There is a powerful tendency to focus on the highly visible mainline components of the software, and to underestimate or completely miss the unobtrusive components (e.g., help message processing, error processing, and moving data around) {49} (p125).

- As the project develops, the "Undiscovered Job Tasks" are progressively discovered as "the level of knowledge we (TAH&SEM) have of what the software is intended to do (increases)" {49} (p126).

- As the additional tasks are discovered, they are then incorporated into the project – incorporated into the project's Work Breakdown Structure, the Gantt and/or PERT charts, or the Earned value system (p126).

- When additional tasks are discovered in a project, they do *not* necessarily trigger an adjustment to the project's man-days estimate {49}. Only if the additional tasks are perceived as requiring a *relatively* "significant" amount of effort to handle would project members "bother" to go through the trouble of formally developing cost estimates and incorporating them in project's work plan (p126).

- If the relative size of discovered tasks is lower than a threshold, the newly discovered tasks are totally absorbed without triggering any adjustments to the project's man-days estimate. If, however, the relative size exceeds the threshold value, part or all of the additional tasks are translated into additional man-days in the project's plan (p127).

2.12 Chapter 11: Planning

- In the planning subsystem initial project estimates are made to start the project, and then those estimates are revised as necessary throughout the project's life (p129).

- The weighting factor WCWF is termed the "Willingness to Change Work Force Level". It is a variable that assumes values between 0 and 1, inclusive. When WCWF = 1, the weighting considers only the "Indicated Work Force Level"; management is adjusting its work force level to the number perceived required to finish on schedule. As WCWF move towards 0, more and more weighting is given to the stability of the work force. When WCWF equals 0, the weighted number of employees desired is wholly dependent on work force stability. In some cases the management becomes increasingly willing to "pay any price" necessary to avoid overshooting the "Maximum Tolerable Completion Date". In such cases, management is often willing to hire more people (p131).

- It is important to realise that the variable WCWF is an expression of a *policy* for managing projects. A range of functions is possible here, representing different strategies on how to balance work force and schedule adjustments throughout the project to minimise overruns and costs (p134).

2.13 Chapter 12: Case study

- The objective of the case study is to examine the model's ability to reproduce the dynamic patterns of a completed software project (p139).

- The case study was conducted at the Systems Development Section of NASA's Goddard Space Flight Center (GSFC) at Greenbelt, Maryland. This organisation is engaged in the development of application software that supports ground-based spacecraft attitude determination and control (p139).

- The requirements for the DE-A project (DE = Dynamics Explorer) were to design, implement, and test a software system that would process telemetry data and provide definitive attitude determination as well as real-time attitude determination and control support for NASA's DE-A satellite (p139).

- The DE-A project was selected for the case study by NASA to satisfy three criteria: it had to be (1) medium in size (i.e., 16-64 KDSI), (2) recent, and (3) "typical," i.e., developed in a familiar in-house software development environment (p140).

- The life cycle phases covered in the case study include the design, coding, and system testing phases. Excluded from the study are the requirements definition phase and the acceptance testing phase because they both lie outside the boundary of TAH&SEM's model, and the requirements phase had not been included in the group's project responsibility. Requirements were instead the responsibility of the user organisation (p140).

- The development and target operations machines were the IBM S/360-95 and -75. The programming language was mostly FORTRAN (85%); assembler language and assembler language macros constituted the remaining 15%. The size of the system in Delivered Source Instructions (DSI) is 24,400 DSI (p140).

- The project's actual key development dates were: design (Oct.1, 1979 - May 9, 1980), coding (May 10, 1980 - March 27, 1981) and sys. test (Nov. 15, 1980 - April 24, 1981). Thus, the project was completed in 19 calendar months. The project consumed 2,222 man-days of effort (2,784 man-days were expended to complete the total project, of which 562 man-days were consumed in the acceptance testing activity) (p140).

Remarks

- the "Hiring Delay" (30 days) is somewhat lower than the industry (40 days) (p141);
- at the System Development Section 25% of an experienced employee's time is committed per new employee (p142);
- the "Average Assimilation Delay" was set for the DE-A project at 20 working days. This value is much lower than values reported in the literature (p142);
- the "Nominal Potential Productivity" is the set of factors that affect productivity that tend to remain invariant during the life cycle of a single project (p142);
- actual productivity rarely equals potential productivity because of losses from communication and motivation problems (p142);
- the DE-A team size was approximately 10 people during most of the development period (p143);
- the nominal error rate was defined to be that generated by the average experienced employee (p143);
- because NASA's launch of the DE-A satellite was tied to the completion of the DE-A software, serious schedule slippages were not tolerated (p144);

- the DE-A Software Project was initiated on October 1, 1979 and the DE-A satellite's launch date was August 3, 1981. All software had to be accepted and frozen by May 3, 1981 (p144);
- the actual DE-A project size was 24.4 KDSI. At the initiation of the design phase, the project's size was under-estimated by 35% (p144);
- it is important for the manager to use a model that is tuned to the specific environment and corresponds well with the resources expended for similar past projects. Managers must use the results of the model, together with historical knowledge of similar systems, to update resource and cost estimates (p145);
- the dynamic behaviour of management systems tends to be largely a function of the interaction of policies that govern such systems {101} (p147);
- the model accurately portrays management's inclination not to adjust the project's scheduled completion date during most of the development phase. Adjustments are instead made in the project's work force level. This behaviour pattern arises, according to DeMarco, for political reasons (p147);
- adjustments in the project's man-days budget start toward the end of the design phase (p147);
- significant adjustments in both the project's man-days and the schedule continue to be made until the final stages of development, an outcome that the model successfully reproduces (p148);
- lower manpower levels mean lower communication and training overheads, which mean a slight over-estimation of productivity (p148);
- with more people at hand in the actual project a smaller schedule overshoot was achieved (p149);
- the work force level shot upward toward the end of the project because of NASA's tight scheduling constraints. As explained earlier, serious schedule slippages were not tolerated (p150);

Summary of NASA DE-A Case Study

- The objective of the case study was to test the model's ability to reproduce the dynamic behaviour of a completed software project. TAH&SEM first parameterised the model by setting model parameters that reflected the particular DE-A project environment. The parameters were obtained from two sources: interviews at NASA and project documentation. The model parameters did *not* involve any changes in the formulation of the model's policy structures. The parameter set merely defines the (DE-A) environment within which the policies were exercised. This is significant since the dynamic behaviour is largely a result of the interaction of the model's policy structures, which were unchanged (p151).

- Four DE-A project variables were examined: completion date estimates, man-day estimates, cost in man-days, and work force loading. While the model was quite accurate in reproducing the project's patterns of dynamic behaviour, it slightly underestimated the absolute value of DE-A's work force level. DE-A's management was more aggressive in its manpower acquisition policy than the model assumed. The underestimate caused the model to underestimate by 6% the project's cost in man-days and to overestimate by 2% the project's duration (p151).

What-if questions

One advantage of system dynamics modelling is that it allows us not only to generate the dynamic implications of a given set of policies but to explore the implications of new and different sets of managerial policies and procedures.
Let us ask some of the "what-if" questions that DE-A's management might ask (p151):

1. What if a different estimation tool was used? In the DE-A project, estimation by NASA's Meta-Model was used and adjusted by management's experience and judgement. Like NASA, other software development organisations have developed quantitative software estimation tools, e.g., TRW's COCOMO model. How can the applicability of such new tools to the NASA environment

be evaluated? To what extent are such models portable to the NASA environment? If not, why not? How can the relevancy of new estimation models be improved?

2. What if more or less quality assurance (QA) effort was expended? In the DE-A project 30-40% of the development effort was allocated to QA, a level that is significantly higher than the industry average. Is this an "optimal" allocation? How can we (TAH&SEM) determine what an "optimal" allocation is? What project/organisational factors affect such a determination?

3. What if more people had not been added at the final stages of the project? Brooks' Law suggests that adding more people to a late project makes its completion date later. When would the DE-A project have been completed had management resisted adding more people at DE-A's final stages?

2.14 Chapter 13: Model Behaviour

- A system dynamics model is a tool that allows repeated experimentation with the system, testing assumptions or altering management policies. The purpose is to gain an understanding of and make predictions about the implications of managerial actions, policies and procedures (p153).

- The most important advantage of a simulation model is its ability to "play out" the dynamic consequences of a given set of assumptions in a way the human mind can do neither well nor consistently; a useful model produces scenarios that are both realistic and explainable in the policymaker's own terminology (p153).

- Using the system dynamics model as an experimentation vehicle can provide many benefits to the software engineering community (p153).

- In software engineering it is remarkably easy to propose hypotheses and remarkably difficult to test them. Accordingly, it is useful to seek methods for testing software engineering hypotheses {265} (p153).

- Unfortunately, controlled experiments in the area of software development tend to be costly and time-consuming {190} (p153).

- In addition to permitting less costly and less time consuming experimentation, simulation models make perfectly controlled experiments possible (p153).

- By using a model of a complex system, more can be learned about internal interactions than would ever be possible through manipulation of the real system (p154).

- EXAMPLE is the prototype project for the experiments. TAH&SEM run the model to simulate the EXAMPLE project and observe and analyse the behaviour of several significant project variables (work force level, schedule completion time, errors and productivity) (p154).

- Barry Boehm's excellent book, *Software Engineering Economics*, provides a wealth of data on the software production environment TRW (p154).

- TAH&SEM draw on projects Boehm described as "the most common type of software project: the small-to-medium size (project) developed in a familiar, in-house, organic software development environment (p155).

- Actual productivity rarely equals potential productivity because of losses from communication and motivation problems (p155).

- A full-time employee allocates, on the average, 60% of his or her time to productive work on the project (p154).

- Boehm notes that "The software undersizing problem is the most critical road block to accurate software cost estimation". This is substantiated by the experiences of several other authors {61,78,81,85} (p156).

- Most software projects start with a smaller core of designers, and as the project develops, the work force slowly builds up to higher levels (p157).

- The early phase of development constitutes the architectural design phase; the emphasis is on determining the overall structure of the system, decomposing the system into its major components, and specifying the interfaces between the components {109}. Many implementation details such as help message processing or error processing would still not be visible. The rate of discovering such necessary job tasks remains low. The rate however, starts to accelerate rapidly as the project work moves into the detailed design phase (p157).

- When the "Job's Size in Man-Days" is adjusted upwards, it is done by adjusting the project's work force level, the project's schedule completion date or both. The choice is really an expression of management's policy on how to balance work force and schedule adjustments throughout the project (p160).

- In the early stages of a project, project managers are generally willing to adjust the work force level to maintain the project on its scheduled course. However, as the project proceeds, management becomes increasingly reluctant to add new people out of an increasing desire that the work force stabilise (p160).

- The shift away from work force adjustments to schedule adjustments continues as the project progresses (p160).

- The reason the work force level in the NASA project shoots upwards towards the end of the project has to do with NASA's tight scheduling constraints. Because software is embedded in a large and expensive space system, serious schedule slippages cannot be tolerated. Management becomes increasingly willing to "pay any price" to avoid overshooting the "Maximum Tolerable Completion Date" even if they must hire more people (p162).

- Non-productive time accounts for about 40% of the software person's time on the job (p162).

- Software developers tend to work harder by allocating more man-hours to the project in an attempt to compensate for the perceived shortage and bring the project back on schedule. They compress their slack time, and then, if necessary, work overtime. The man-hours lost per-day decreases (p163).

- The two "spikes" in overwork occur as an *explicit* project milestone is approached. The first spike occurs towards the end of the development phase, and the second spike occurs towards the end of the only other explicit milestone in the model, the end of the system testing phase. This pattern was observed by Boehm and labelled the "Deadline Effect" phenomenon (p164).

- Project members are not willing to maintain an above-normal work rate indefinitely. Once people start working at a rate above their normal rate, their "Overwork Duration Threshold" decreases because people enjoy and need their slack time (p164).

- As a result of NASA's exceptionally high expenditures on Quality Assurance, in the NASA project a larger fraction of the errors is detected early in the development phase when errors are less costly to detect and correct (p166).

2.15 Chapter 14: On the accuracy of software estimation

- For many years, estimation of software project development time and cost had been an intuitive process based on experience and analogy {200} (p167).

- An empirical model uses data from previous projects to evaluate a current project. A theoretical model uses formulae based on global assumptions, such as the rate at which people solve problems or the number of problems available for solutions at a given time (p167).

- Even today, almost no model can estimate the true cost of software with any degree of accuracy {130} (p167).

- The serious student of estimating must first be willing to probe deeply into the fascinating and complex system development process, to uncover the phases and functions of the process (p168).

- Having captured within the integrative system dynamics model "influence variables of software development and their causal relationships", TAH&SEM embark on a quantitative analysis of software cost and schedule estimation (p168).

- By imposing different estimates on a software project different projects will be created. Research clearly indicates that the decisions that people make in projects and the actions they take are significantly influenced by the pressures and perceptions produced by the project's schedule {57,109,125,224,236} (p169).

- A higher work force level generally means more communication and training overhead, which in turn affect productivity negatively. Scheduling can dramatically change the manpower loading throughout the life of a project (p170).

- When the project is perceived to be ahead of schedule, "Parkinson's Law indicates that people will use the extra time for ... personal activities, catching up on the mail, etc." {49}. This, of course, means that they become less productive (p170).

- In addition to permitting less costly and less time consuming experimentation, simulation experimentation makes perfectly controlled experiments possible {99} (p170).

- The management reserves ranged from 5% to 50% of the estimated software cost with a mean of 18% {85} (p171).

- The safety factor is simply a mechanism to bring the initial man-days estimate closer to the project's true size in man-days. The larger the safety factor the smaller the estimation error (p172).

- The project's initial estimates create pressures and perceptions that affect how people behave on the project. Overestimating a project often leads to an expansion of the project members' slack time activities, which leads to further reductions in productivity (p173).

- When excesses are small, they tend to be absorbed as expanded slack activities. However, there is a limit on how much "fat" employees are willing or allowed to absorb. Beyond those limits man-day excesses become translated into cuts in the project's work force, schedule, or both (p174).

- The "Safety Factor Policy" does achieve its intended objective: more accurate estimates. However, the organisation pays dearly for this (p175).

2.16 Chapter 15: Portability of estimation models

- Many different kinds of quantitative software estimation models have been developed. In almost all cases the model is based either directly or indirectly on historical data {236}. Another approach is to formulate a parametric model, a mathematical function of several variables suggested by previous experimentation and engineering judgment (p177).

- Even today, almost no model can estimate the true cost of software with any degree of accuracy {184} (p177).

- Portability of the models has proven to be especially poor {43,49,144,184} (p177).

- Both the accuracy and the portability of software estimation models can be significantly improved by taking into consideration not only the technical aspects of the software development environment, as is the case with the current models, but also the managerial and organisational characteristics of the environment (p177).

- Mohanty made his hypothetical software project 36,000 machine-language executable instructions. The 12 cost estimates varies from a low of $ 362,500 (the Farr and Zagorski Model) to a high of $ 2,766,667 (the Kustanowitz Model) for the *same* software project. In other words, the highest estimate is over 650% higher than the lowest (p178).

- Two sources of variation were suggested by Mohanty: quality of the final product and the company's environment reflected by the model {184} (p178).

- According to Boehm and Wolverton, software cost estimation models should include only objective variables to avoid allocating the software cost variance to poorly calibrated subjective factors such as complexity. The inclusion only of objective variables makes it harder to jiggle the model to obtain any result that one wants. Furthermore, a software cost estimation model should avoid the use of variables whose values cannot be determined until the project is complete (p178).

- The results of a simulation experiment conducted to quantify the impact of four managerial variables on the cost of software development will be discussed. Two of the variables address manpower-acquisition and staffing policy issues while the other two concern issues of effort distribution among the software development activities. The result is significant: based on just these four managerial variables, the cost of the project EXAMPLE varies by a factor of two (p180).

- The policy of allocating project members half-time to the project results in a cost that is 22% higher (project EXAMPLE) (p180). More time is lost on human communication, for example, to resolve questions about design and testing {244}. Moreover, the amount of work itself usually increases in the form of more documentation, modules, and interfaces {69,109}. An increase in training overhead accounts for 10% of the increase (p181).

- Different policies affect what the project's cost will end up being *and should therefore be explicitly considered when project cost estimates are made* (p182).

- More people on the project means more work gets done. It also means that the project team's overall productivity is lower because of the increased communication and training overheads (p184).

- Variables used in cost estimation tend to be those which are easier to measure, quantify, and estimate, even if they are not the most significant {65} (p184).

- The results may seem to contradict Brooks' Law, which states that "Adding manpower to a late software project makes it later" {57}. TAH&SEM's model results indicate "adding manpower to a late software project makes it more costly, but *not necessarily later*" (p184).

- A major conclusion is that TAH&SEM do not possess an adequate understanding of resource consumption behaviour over the life cycle development phases {172} (p184).

- A different distribution of estimated effort among a project's phases creates a different project (p185).

- Re-run the project EXAMPLE with the following four adjustments:
    - Set the value of the "Average Daily Manpower per Staff" at 0.5 (the base case value was 1);
    - The "Willingness to Change Work Force" is formulated in terms of the "Hiring Delay", yielding a more aggressive manpower acquisition policy (in the base case it is formulated in terms of the (Hiring Delay + Average Assimilation Delay));
    - Allocation of effort among the development and testing phases is set at 60% development and 40% testing (in the base case it is 80-20);
    - The "Planned Fraction of Manpower for QA" is set at 20% (in the base case it is 15%).

  The result of running EXAMPLE with this different set of managerial policies is a total cost of 7,316 man-days, *almost double the base case cost of 3.795 man-days* (p187).

2.17 Chapter 16: Analogy method of software estimation

- Estimation by analogy involves reasoning by analogy with one or more completed projects to relate their actual costs to an estimate of the cost of a similar new project {49} (p189).

- The most common technique on making operational estimates is the use of experience gained on one or more similar projects {200} (p189).

- A project's estimate creates pressures and perceptions that directly influence the decisions people make and the actions they take throughout the project's life cycle (p189).

- There appear to be inherent factors in the management of a software project that would cause it to overrun even with a "perfect" schedule estimate (p192).

- Because of the inherent tendency to overshoot, the use of the analogy method in estimating injects a bias in scheduling, a bias that in the long-run generates longer than necessary schedules (p192).

- Through further experimentation with the model it was possible (in relation to project EXAMPLE2) to isolate the real cause of this persisting schedule-overrun problem: the interaction of two factors, the manpower-acquisition policy and the turnover of project personnel (p192). Any drop in the work force level because of turnover will in turn decrease the value of the man-days remaining, creating a scheduling problem (p193).

- The phenomenon of projects consuming longer and longer schedules is one that has been frequently encountered in system dynamics studies of organisational behaviour {241} (p193).

- When a software development project overruns its schedule there is an apparent cause: the project was poorly estimated (p194).

- After all, software estimation is not yet an exact science. Significantly, it is often impossible in a real life situation to demonstrate that underestimation was *not* in fact the cause (p194).

2.18 Chapter 17: The 90% syndrome

- There is ample evidence in the literature to support the pervasiveness of the "90% syndrome" in the management of software development projects {28,81,85,89,242} (p197).

- Baber provides the following description of the problem:
  "... estimates of the fraction of work completed (increase) as originally planned until a level of about 80-90% is reached. The programmer's individual estimates then increase only very slowly until the task is actually completed" {28} (p197).

- Progress generally lacks visibility in the earlier phases of development; it is measured by the rate of expenditure of resources rather than by of actual accomplishments, and status reporting ends up being nothing more than an echo of the project's plan. This creates the "illusion" that the project is on target. However, as the project approaches its final stages (when 80-90% of the resources are consumed), discrepancies between the percentage of tasks accomplished and the percentage of resources expended become increasingly apparent. At the same time, project members become increasingly able to perceive how productive the work force has actually been (p198).

- When *man-days* requirements are underestimated the problem often remains undetected until the final stages when most of the budgeted man-days are consumed (p198).

- The "90% syndrome" arises because of the interaction of two factors: underestimation and imprecise measurement of progress (p199).

- "A surrogate is a substitute measure of some phenomenon that is used because it is not feasible to measure the phenomenon directly" {22}. For software, consumption of resources is the surrogate often used to measure progress (p199).

- The better the measurement tool the earlier it will detect that progress is not keeping up with the underestimated schedule (p200).

- When a discrepancy is detected early in the development cycle, management usually reacts by adding more people rather than by adjusting the schedule. This happens, according to DeMarco, for political reasons: e.g., "It's too early to show slip. If I re-estimate now, I risk having to do it again later (and looking bad twice)" (p201).

- The result of sticking with a schedule that is too tight is often an increase in the project's cost {49} due to a large work force level (p202).

- Since the consequences of a decision often occur much later than the decision itself, it is difficult for the members to trace backward from the disruptive consequences to determine precisely what caused them. The members cannot make such an analysis, simply because there are too many competing explanations. Thus the only thing members can do when a new problem arises is to engage in more localised problem-solving {259} (p202).

- Even though the issues TAH&SEM are raising here on the dysfunctional consequences of measurement tools are beyond the scope of this book, TAH&SEM feel that a general integrative approach provides the viable basis for addressing them (p202).

2.19 Chapter 18: The economics of quality assurance

- Quality Assurance (QA) is a set of activities performed in the development of a software product to reduce doubts and risks about the performance of the product in the target environment {210} (p203).

- Software quality assurance is approached by two distinct and complementary methodologies:
  - the first is to assure that the quality is initially built into the product. This methodology emphasises that a coherent, complete, unambiguous, and no conflicting set of requirements be designed early in the project;
  - the second is to review and test the product as the product is designed and coded {84} (p203).

- TAH&SEM model's development phase includes both design and coding activities but *excludes* the development of the requirements. Thus, TAH&SEM will focus on the second QA methodology (p203).

- Several specific techniques are available for reviewing and testing the software product as it is designed and coded:
  - structured walk throughs and technical reviews {105};
  - inspections {94};
  - code reading (by a programmer other than the original designer) {260};
  - integration testing {172} (p203).

- In this chapter TAH&SEM focus on economics rather than on technical aspects of QA. TAH&SEM investigate the trade-off between the benefits and costs of QA in terms of the total project cost (p203).

- A pressing concern to the software quality manager is how cost efficient are the QA operations during the development cycle {146} (p204).

- There are no published studies investigating "how cost efficient are the QA operations during the development cycle" (p204).

- The question is not whether QA is justified but *how much* of it is justified (p204).

- The primary goal of QA is "that errors be detected and corrected as early as possible and only a minimal amount of problems be allowed to slip from one phase of the development to the next" {256} (p204).

- McClure reports that Shooman determined that detecting and correcting a design error during the design phase required *one-tenth the effort* that would have been needed to detect and correct it during the system testing phase because of the additional inventory of specifications, code, user and maintenance manuals, and so on that would also have required correction {166} (p204).

- A significant feature of the relationship between the QA effort expended and the percentage of errors detected during development, is the diminishing returns as QA expenditures exceed 20-30% of development effort (p204).

- QA is used not only in the development phase but also to minimise the cost of the testing phase (p208).

- QA policy does have a significant impact on total project cost (p206).

- Although the optimal QA percentage is dependent on the environment (e.g., varying from 11% to 20% in the experiments), it is possible to represent the characteristics of the environment in the model to ascertain the appropriate QA allocation for a specific project (p210).

2.20 Chapter 19: Model enhancement and Brooks' Law

- TAH&SEM's model has increased their understanding of and enabled them to make predictions about the management of software development and established the viability of the system dynamics methodology as an effective research vehicle (p211).

- TAH&SEM's objective in this chapter is to illustrate this process by examining a limited model enhancement by investigating Brooks' Law (p211).

- Brooks' Law was first publicised in Dr. Fred Brooks' 1975 book, *The Mythical Man-Month: Essays on Software Engineering* (p211).

- Brooks' Law is stated as follows: "*Adding manpower to a late software project makes it later*" {57}. The lack of interchange ability between men and months was recognised by Brooks as being caused by training and intercommunication overheads (p212).

- Intercommunication is worse. If each part of the task must be separately coordinated with each other, the effort increases as n(n-1)/2. Three workers require three times as much pair wise intercommunication as two; four require six times as much as two ... (p212).

- Management's policy on how to balance work force and schedule adjustments is represented in the model by "Willingness to Change Work Force". By adjusting this variable TAH&SEM can examine the impact of more aggressive manpower acquisition policies on the project's cost and duration (p218).

- TAH&SEM's experimental results provide insight into Brooks' Law. Adding more people to a late project always causes it to become more costly, but it does *not always* cause it to be completed later. The increase in the cost of the project is caused by the increased training and communication overheads, which in effect decrease the average productivity of the workforce and increase the project's man-day requirements (p222).

2.21 Chapter 20: Conclusions and future directions

- The objective of this book is to enhance the understanding of and gain insight into the general process by which software development is managed (p223).

- TAH&SEM developed an integrative system dynamics model of software development project management, conducted a case study to test the model, and used the model to study and predict the dynamic implications of an array of managerial policies and procedures (p223).

- An integrated approach helps TAH&SEM achieve an overall understanding (p223).

- The schedule overshoot problem can arise not only because of schedule underestimation, but also because of management's hiring policies (p223).

- Managerial intervention often leads to second- and third-order consequences. TAH&SEM's analysis of the "Safety Factor Policy" in scheduling software projects showed that while such a policy succeeds in producing more accurate project estimates, it tends to "create" more costly projects (p223).

- The model identifies feedback mechanisms and uses them to structure and clarify relationships in software project management (p223).

- The mathematical formulation of a system dynamics model requires that the structural relationships between variables be explicitly and precisely defined. As such, the model lays the foundation for the theory of software project management (p224).

- The objective of the case-study was to examine the model's ability to reproduce the dynamic behaviour of a completed software project (p224).

- If "understanding" is the intellectual outcome of a theoretical model, then "prediction" is its practical outcome {90} (p224).

- A different schedule creates a different project (p225).

- An important implication is that a software estimation model cannot be judged solely based on how accurately it estimates historical projects (p225).

- Evidence in the literature indicates that currently available quantitative software estimation tools are not particularly portable from the company in which they were developed to another (e.g., see {43,49} (p225).

- TAH&SEM identified four aspects of a company's managerial environment (manpower acquisition, manpower allocation, effort distribution and QA allocation) (p225).

- The experiment generated two interesting insights:
  - first, that there are inherent factors in the management of a software project (resulting from the interaction of manpower acquisition policies and personnel turnover) that cause it to over-run even what would amount to be a "perfect" schedule estimate;
  - second, that because of the inherent tendency to overshoot, the use of the analogy method injects a bias in the scheduling process, a bias that generates longer than necessary project schedules (p225).

- Two sets of experiments were conducted on QA. The objective of the first was to investigate not whether but *how much* QA is justified. To do this TAH&SEM first examined the relationship between QA expended and the percentage of errors detected during development. A significant feature of this relationship is the "diminishing returns" of QA as QA expenditures extend beyond

20-30% of development effort. TAH&SEM then derived the "optimal" QA expenditure level, the level that minimises total project cost (p225).

- The objective of the second set of QA experiments was to examine the sensitivity of the results to two project variables: the distribution of effort between the development and testing phases, and software development productivity. The findings constitute "rules-of-thumb" that organisations can use to adapt published results from other organisations to their own environment (p226).

- The experimental results do *not* support Brooks' Law for the type of project studied in TAH&SEM's research. Adding more people to a late project causes it to become more costly but not to be completed later unless hiring continues to the end of the project's testing phase. Brooks' Law does not universally apply to all software development environments (p226).

- TAH&SEM believe that their work has pointed up several areas requiring more intensive research:
    - *Model Enhancement* (requirements definition/analysis phase; multiple projects; large-scale project environments; quality of the produced software product);
    - *New Modelling Applications* (the modelling approach described here can be extended to investigate a broader set of issues pertaining to the software development *organisation*) (p227).

APPENDIX 3

THE SÓXIMA MODEL

Introduction

During the audits at FINANCE, it turned out that it was not always clear what information had to be gathered within the framework of SOX. Therefore, it is advisable to arrive at a detailed SOX information architecture for FINANCE in order to be able to comply with SOX and for demonstrating that FINANCE is in control. In this appendix I have shown this by using the Sóxima model [Van Dijk 2006]. Sóxima stands for: SOX Information Management and Architecture.

Tbe information needed for being able to meet the SOX requirements can be collected just before the report period but at that moment that often proves a difficult and time-consuming job. Furthermore, it often then transpires that not all of the necessary SOX information is available. For that reason, it is preferable to make production and collection of the required SOX information part of the daily processes.

SOX information has a certain coherence (information architecture, as mentioned in sheet 1) and relates to processes, changes/projects and activities (see sheet 2).
Sheet 3 conveys the usual IPO scheme (Input Process/Change Output). SOX information can be obtained by defining for each process/cbange which information is relevant for SOX (see sheet 4).

Within FINANCE, one works with Business Units. Business Unit BU1 has divisions which include the division IT Infrastructure. Tbis division includes amongst other things the sub divisions Infra1 and Infra2 (see sheet 5). Within FINANCE a number of processes are in progress (including the processes A, B, C, D , E and F, see sheet 6) and a number of changes/projects are in process (including the cbanges A, B, C, D, E and F, see sheet 7). Within processes, activities are carried out; process K for example carries out the activities PK1, PK2, PK3, ... (see sheet 8). Activities are also carried out for the henefit of cbanges; for change G for example, the activities CG1, CG2, CG3, ... are carried out (see sheet 9).

As an example, we examine sub division Infra1. In which activities is Infra1 involved? Sheet 10 indicates that Infra1 is involved in the activities PK3 and PK5 of process K and in the activities CG1, CG7 and CG11 of change G. Which SOX information does Infra1 have to collect in this respect? Sheet 11 gives an example of the relationship of the activities in question with the corresponding SOX information types/elements. Activity PK5 is for example related to the following SOX information types: type 2, type 3 and type n. The matrix in sbeet 11 sbows all rclevant activities and for each activity the applicable SOX information types. This lays down which SOX information has to be delivered by sub division Infra1. In order to ensure that the right information is collected at the rigbt time, a Management Control System is necessary. Sheet12 gives an example of a general Management Control System (MCS). This needs to be set up for every (sub)division. By allowing the MCSs to function properly, collection of the required SOX information is realised within FINANCE as part of the daily routine/processes and the required SOX information can bc delivered at any given time.

Sóxima

SOX Information
Management and Architecture

November 2006 - Sóxima – Aart van Dijk

Sheet 1: Sóxima: SOX Information Management and Architecture

Sóxima

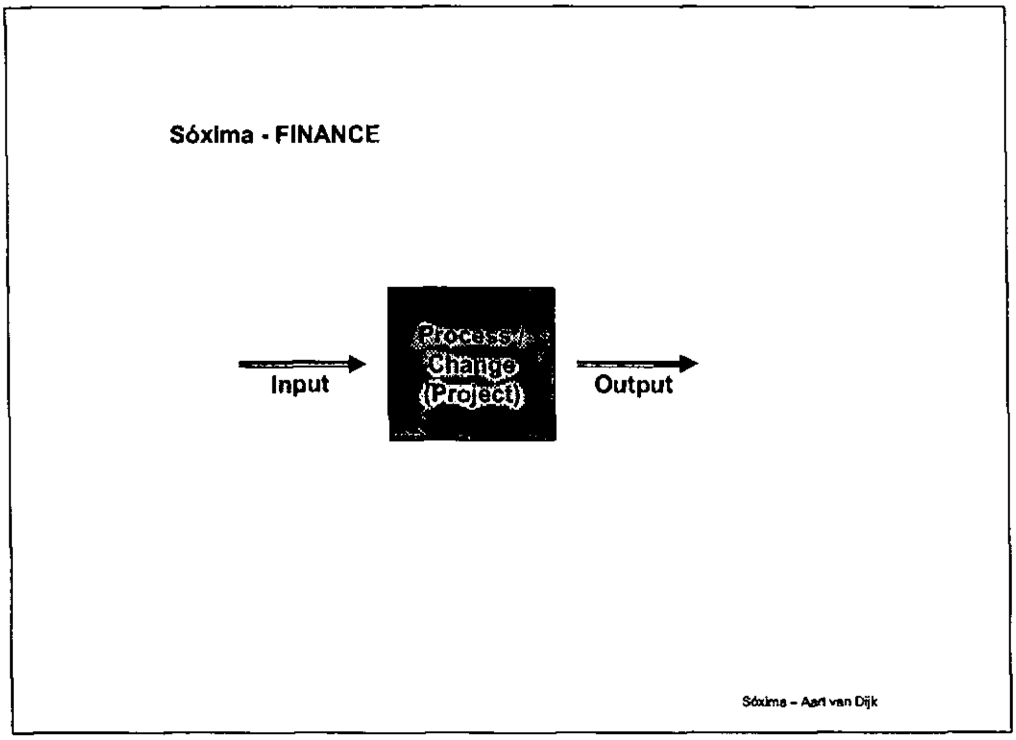
- Information
- Processes
- Changes (Projects)
- Activities

Process: a (well-)ordered collection of (process-)activities

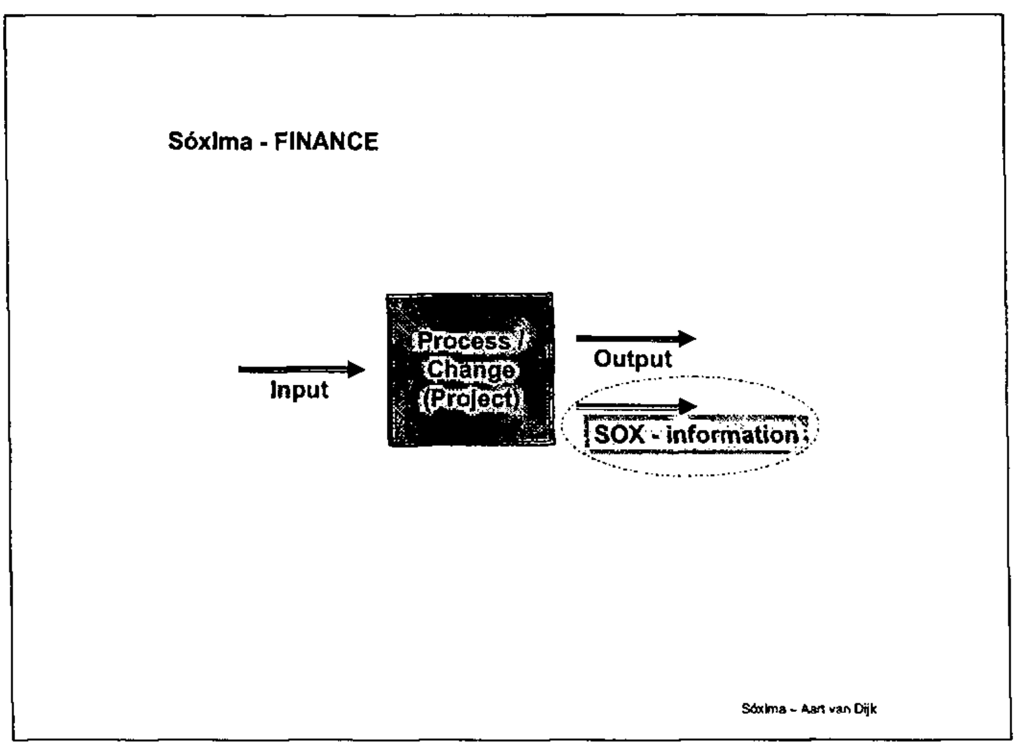
Change (Project): a collection of single use activities

Sóxima – Aart van Dijk

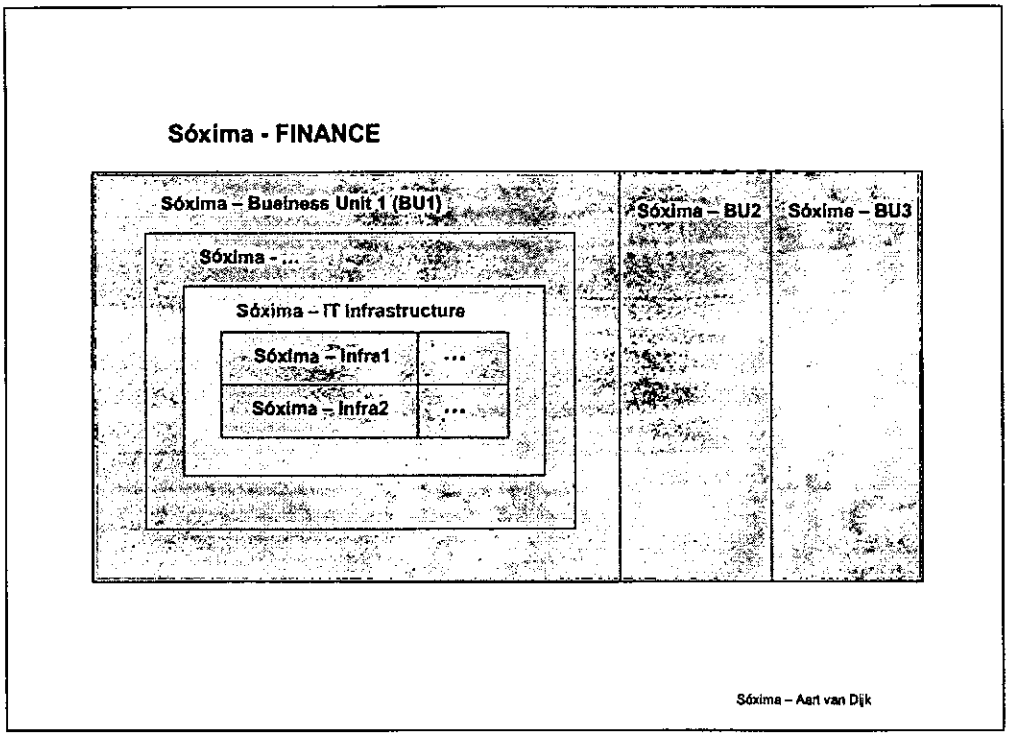
Sheet 2: SOX information concerns processes and process activities with regard to changes/projects



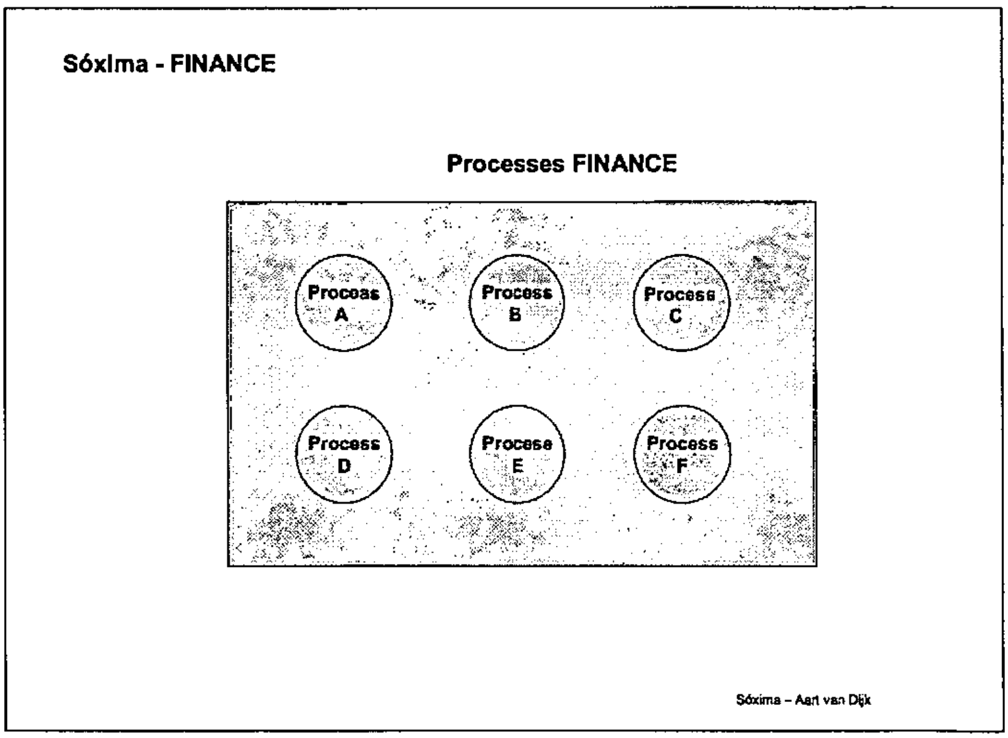
Sheet 3: Standard IPO scheme (Input Process Output)



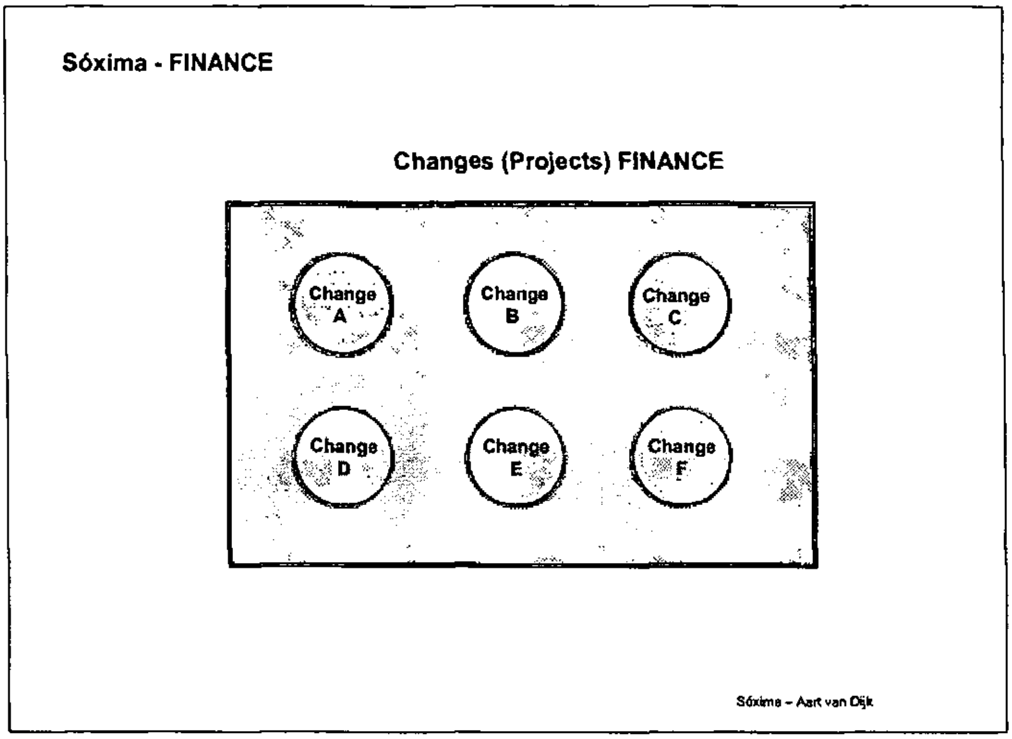
Sheet 4: Extended IPO scheme: Extra information for the benefit of SOX



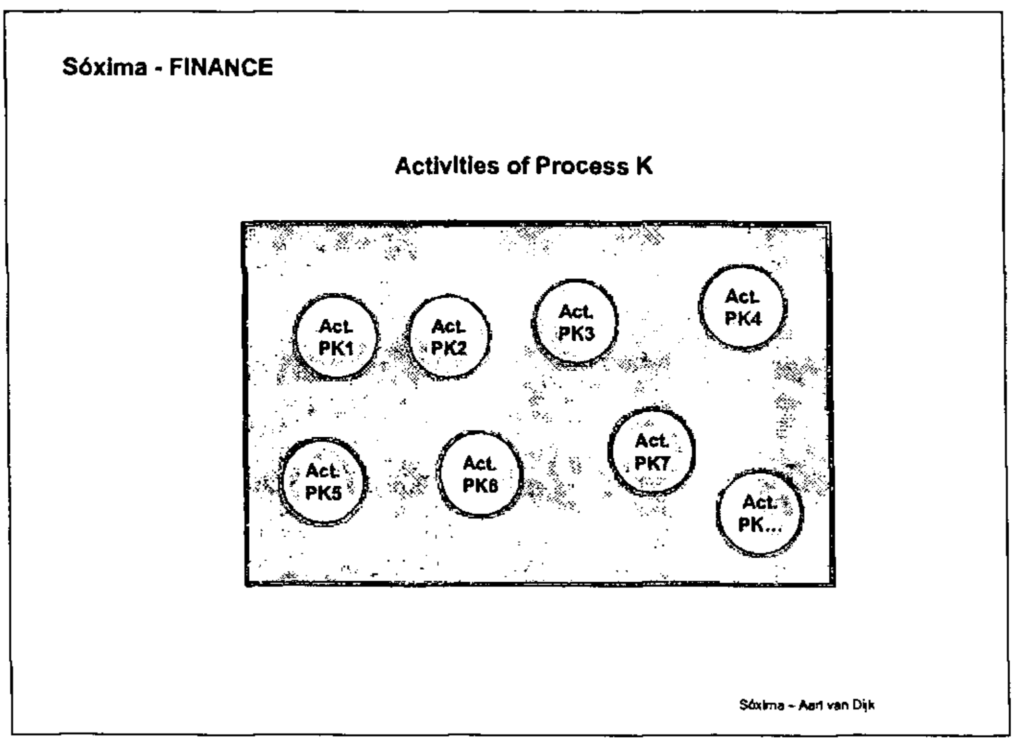
Sheet 5: Division IT Infrastructure as part of Business Unit 1 with sub divisions Infra1 and Infra2



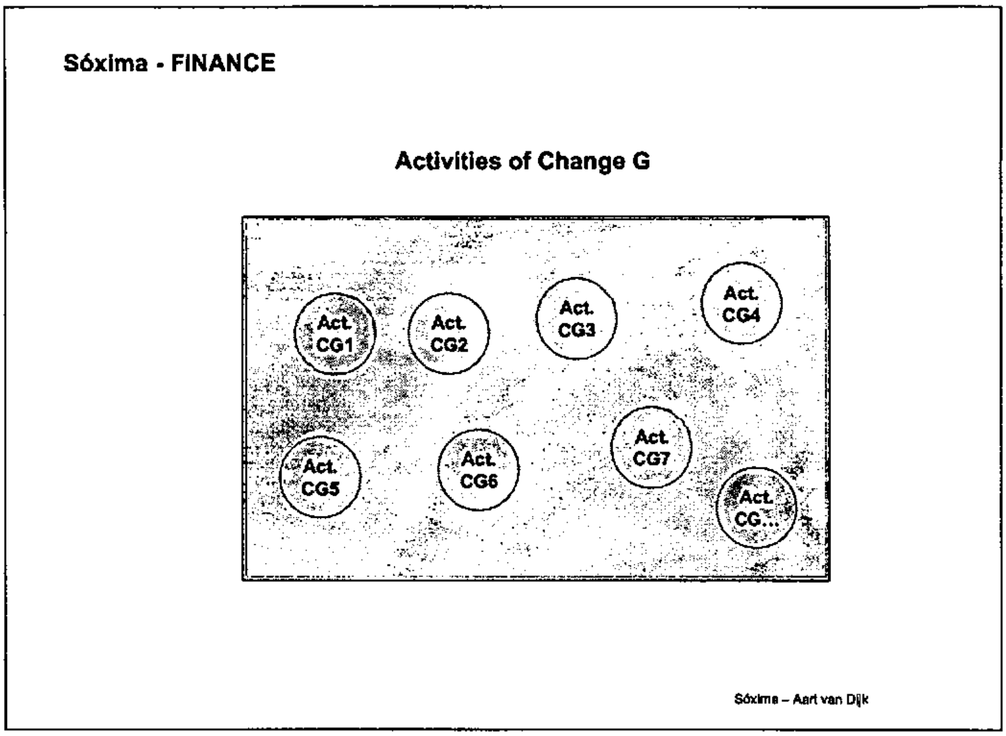
Sheet 6: A few FINANCE processes (A, B, C, D, E and F)



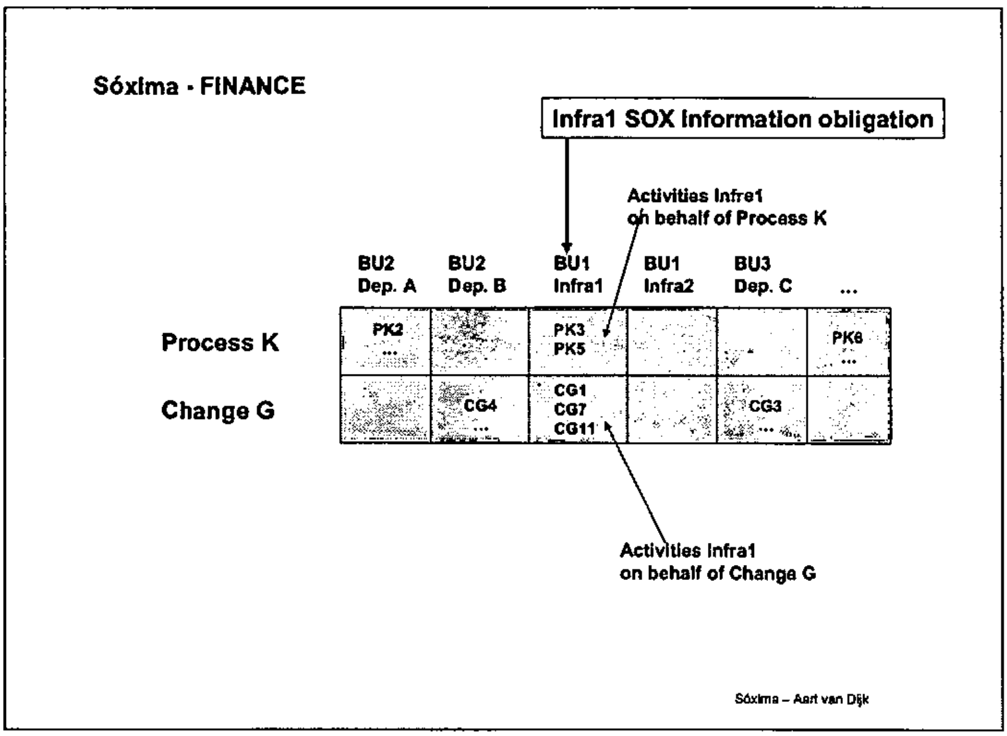
Sheet 7: A few FINANCE changes/projects (A, B, C, D, E and F)



Sheet 8: A few activities of process K (Act. PK1, Act. PK2, ...)



Sheet 9: A few activities of change G (Act. CG1, Act. CG2, ...)



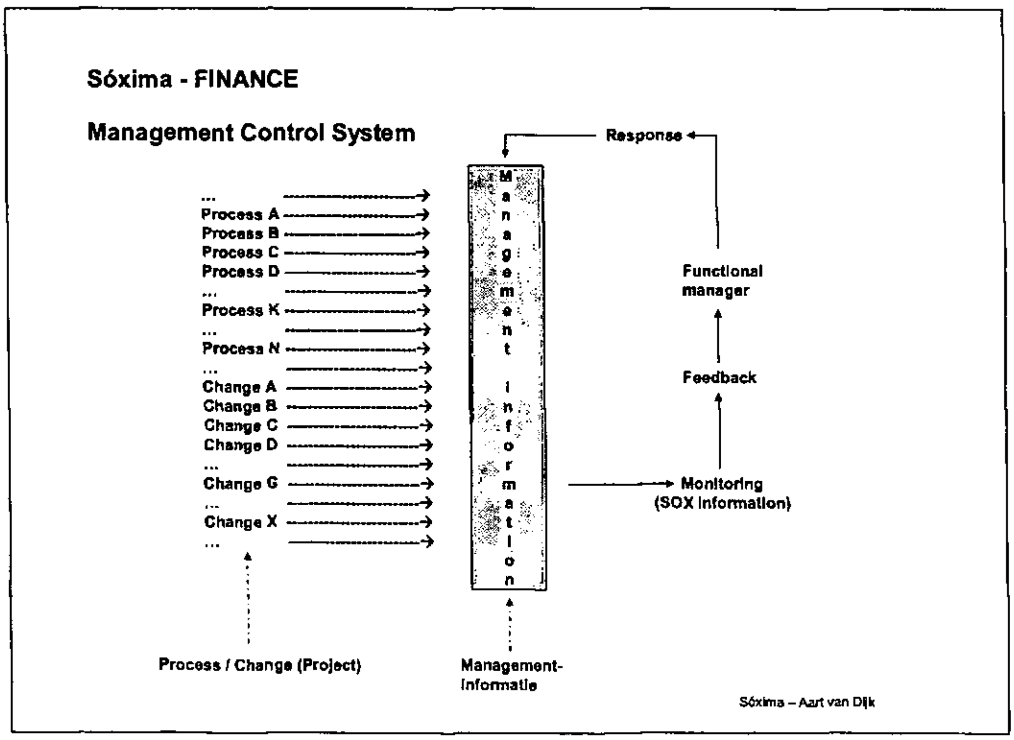
Sheet 10: Sub division Infra1 is to do with process K and change G in particular with the activities PK3, PK5, CG1, CG7 and CG11

Sóxima - FINANCE

Infra1	SOX Information architecture / Inform. types (elements)								
Activities	Type 1	Type 2	Type 3	Type 4	...	...	...	...	Type n
...									
PK3	x			x		x			
PK5		x	x						x
...									
CG1					x				
CG7		x					x		
CG11	x							x	
...									

Sóxima – Aart van Dijk

Sheet 11: SOX Information Architecture of sub division Infra1, each activity relates to a number of types of information/information elements



Sheet 12: Management Control System (MCS) for SOX Information, each sub division has its own MCS

APPENDIX 4

JOURNAL PUBLICATIONS

1 Dijk, Aart J. van and Anthony S. White,
Success and failure factors in ICT projects:
A Dutch perspective,
This article has been submitted to: Software Process: Improvement and Practice (October 2008)

2 Dijk, A.J. van,
Succes- en faalfactoren bij ICT-projecten,
Bekend maar onbemind*,
Informatie, juli/augustus 2008, blz. 40-44.
**) Success and failure factors in ICT projects, well-known but unpopular*

3 Dijk, Aart van and Ton Algra,
Role-based Access Control (RBAC),
Policy and implementation:
KLPD employee authorisation for using SAP R/3 applications,
IT Management Select, 2-2004, September 2004, pp. 31-47

4 Dijk, A.J. van, R.E. Overeem en J.J. Scbijf,
Gegevensuitwisseling (interfacing) van applicaties met bebulp van een message broker:
De applicaties van de Business Unit BedrijfsCommunicatie van KPN Telecom*
IT Management [Select], 2001-1, april 2001, blz. 37-55
**) Interfacing applications by means of a message broker.*
The applications of the Business Unit Business Communication of KPN Telecom

5 Dijk, A.J. van en J. Weber,
EAI en ERP met Baan Open World,
De realisatie van een adapter in de praktijk van de Business Unit BedrijfsCommunicatie van KPN Telecom*,
Software Release Magazine, 2001-4, juni 2001, blz. 30-35
**) EAI and ERP with Baan Open World. The realisation of adapters at KPN Telecom*

6 Dijk, A.J. van,
Doorbelasting van kosten van gebruik en beheer van corporate data:
De basisregistraties bij het Gak/ASZ,*
IT Management [Select], 1998-4, december 1998, blz. 61-78
**) Charging costs of use and management of corporate data.*
The national databases of employees, employers and employments

7 Dijk, A.J. van,
IMPALA,
Een beheersingsmodel voor de bedrijfstelefonie-services*,
Informatie Themanummer 1994, december 1994, blz. 843-855.
**) IMPALA: Management Control System for company Telephony Services*

8 Dijk, A.J. van,
Schermbeeldcommunicatie en autorisatie*,
Informatie, maart 1994, blz. 191-198.
*) *Screen image communication and authorisation*

9 Dijk, A.J. van,
GIRAF,
Een information retrieval systeem voor universiteiten,
hogescholen en andere pluriforme organisaties*,
Informatie, mei 1984, blz. 445-455.
*) *An information retrieval system for Universities,*
Schools of Higher Vocational Education and other multiform organisations

10 Dijk, A.J. van,
Ervaringen met het "ontwikkeltool" ADS/ONLINE bij de ontwikkeling
van een geautomatiseerd information retrieval systeem*,
Informatie, november 1984, blz. 912-927.
*) *Experiences with the development tool ADS/ONLINE at the*
development of a computerised information retrieval system

11 Dijk, A.J. van,
Delftse studenten vragen computer om boeken
uit de bibliotheek*,
DP-Monitor, april 1979
*) *Students Delft University of Technology ask computer for library books*

12 Dijk, A.J. van,
Ervaringen met het COMMAND-level van CICS/VS bij de
ontwikkeling van een geautomatiseerde bibliotheektoepassing*,
Informatie, april 1980, blz. 313-322.
*) *Experiences with COMMAND-level CICS/VS at the*
development of a computerised library application

13 Dijk, A.J. van en D. Mozes,
VERKEERSDATAVERZAMELING,
Een automatisch systeem voor registratie van verkeersdata
en verwerking tot voertuigtrajectoriën*,
Verkeerskunde, juni 1975, blz. 308-314.
*) *Traffic Data Collection: Equipment and a software system which derives*
vehicle trajectories from the detector records on the magnetic tape

14 Dijk, A.J. van,
BIBLIOSYSTEM,
Computerhulp bij het samenstellen van bibliografieën*,
Informatie, oktober 1971, blz. 448-455.
*) *Composing bibliographies by means of a computer*

Publication 1

Success and failure factors in ICT projects: A Dutch perspective

2008

This article has been submitted to:

Software Process: Improvement and Practice

Abstract

This paper examined the success and failure factors in ICT projects. The low success rate of software projects in terms of reliability, meeting due dates and working within assigned budgets is widely recognised and topical. International as well as Dutch publications and the procedures in Tarek Ahdel-Hamid's work on Software Project Management/Dynamics are discussed. A SUFFI Chart (SUFFI = SUccess and Failure Factors in ICT projects) is developed. The management of a portfolio of projects is compared with the SUFFI Chart. A number of Dutch projects with which the first author was directly involved are examined to show how they compare with the factors identified from the literature. These do show considerable correlation between important SUFFI factors and project success. The portfolio consists of 9 ICT projects and 4 ICT project audits. Projects such as SAP, RBAC, EAI, charging method, PABX, financial building administration, information retrieval, book reservations, traffic data collection, introduction of the Internet functionality and SOX, for different companies/organisations (Delft University of Technology, National Police Services Agency, KPN – Dutch Telecom Company, University of Amsterdam, government, banking). This work shows that for a successful project 4 of the 5 most important SUFFIs have to be absent.

Success and failure factors in ICT projects: A Dutch perspective

Aart J. van Dijk[1], *Anthony S. White*[2]
[1]Avédé-Info BV
Ponyweide 79A
2727 HK Zoetermeer
The Netherlands
Email: avede-info@planet.nl
[2]School of Engineering and Information Sciences,
Middlesex University,
The Burroughs,
Hendon
London NW4 4BT

Abstract

This paper examines the success and failure factors in ICT projects. The low success rate of software projects in terms of reliability, meeting due dates and working within assigned budgets is widely recognized and topical. International as well as Dutch publications and the procedures in Tarek Abdel-Hamid's work on Software Project Management/Dynamics are discussed. A SUFFI Chart (SUFFI = SUccess and Failure Factors in ICT projects) is developed. The management of a portfolio of projects is compared with the SUFFI Chart. A number of Dutch projects with which the first author was directly involved are examined to show how they compare with the factors identified from the literature. These do show considerable correlation between important SUFFI factors and project success. The portfolio consists of nine ICT projects and four ICT project audits. Projects such as SAP, RBAC, EAI, charging method, PABX, financial building administration, information retrieval, book reservations, traffic data collection, introduction of the Internet functionality and SOX, for different companies/organizations (Delft University of Technology, National Police Services Agency, KPN – Dutch Telecom Company, University of Amsterdam, government, banking). This work shows that for a successful project 4 of the 5 most important SUFFIs have to be absent.

Keywords

Success/failure factors, SUFFI, Dutch ICT projects, Software Project Dynamics, project management, software audits

1 INTRODUCTION

The low success rate of software projects in terms of reliability, meeting due dates and working within assigned budgets is widely recognised [1], [2], [3]. Successful project management can be related to technical production processes, time scheduling and individual differences in project managers, members and team processes.

Disasters, such as the United Kingdom (UK) Health Service and the London Ambulance Service computerisation, cost in both money and human terms. Progress has been made in the use of System Dynamics methods to describe the operation of software projects. The models of operation of the software development process were described by the successful System Dynamics (SD) models of Abdel-Hamid & Madnick [4], which set up equations relating levels such as the *number of detected errors*, or *the number of reworked errors* and relates them to rates such as *the error detection rate* or *the rework rate.* Significant features of these models included the decision processes. These models were validated against NASA project data for a medium size project and the agreement is strikingly good.

In this paper the concept of the SUFFI Chart (SUFFI = SUccess and Failure Factors in ICT projects) will be developed. In section 3 a portfolio of projects is analysed with the SUFFI Chart.

Relevance
One may ask the question, whether it is relevant to look at success and failure factors in ICT projects. The number of publications on this subject indicates that many researchers have been or are still studying this subject. As early as 1982, Professor Jan Oonincx wrote his book [5] "Why are information systems still failing?" The first author wrote a review of this book 26 years ago:

Although there are many well-functioning information systems, there are plenty of information systems that either function badly or never reach their original target as set during construction, namely being taken into service. This book discusses, grouped in seventeen chapters, a number of aspects that often cause development and introduction of information systems to fail. Information systems, which are set up too ambitiously, too isolated or without proper planning, stand a very large chance of failing. Insufficient involvement of future users in the development of information systems or a passive attitude of the top management also often lead to disappointing results. The book is suitable for students of information science, business administration, economics etc. Although the IT engineer, employed in practice will encounter a number of familiar cases, he will also benefit from reading the book attentively. (Future) users of information systems will largely reap the benefits from studying this book. In this case, the argument "no time" will not wash. It is a very readable book of modest proportions but with very important contents.
25 August 1982 - ir. Aart J. van Dijk

In his inaugural speech at the Rijksuniversiteit Groningen in 2002, Professor Egon Berghout [6] said the following about Jan Oonincx' publication: "This booklet is, without a doubt, based on many years of annoyance preceding 1982, however twenty years later it can be reprinted almost unchanged. Hardly any causes have been removed".

The American "Standish Group" has been involved for 10 years with research into ICT. In their research, they aim to determine and change success and failure factors regarding such projects. Their study, which has been appropriately baptised "*Chaos*" [7], [8], appears every two years. This study also shows that in 2003 only 34% are successful, 51% did not go according to plan but ultimately does lead to some result and 15% of the projects fail completely.

The abovementioned studies and some new Dutch studies (e.g. Nico Beenker [9], Peter Noordam et al. [10] and Netherlands Court of Audit [11]) show that the subject is still very relevant in the year 2008. It may be concluded that the subject of success and failure factors in ICT projects has been in the spotlight for more than 26 years and still is very topical.

2 WHAT IS UNDERSTOOD BY A PROJECT FAILURE?

The main definitions are:

- "the term "failure" refers to projects that are cancelled without completion due to cost or schedule overruns or that run later than planned by more than 25 percent" [12];
- "a definition of a failed project is quite easy – it is a project which does not make the journey from conception through to successful implementation" [1];
- "a successful project satisfies three factors: it complies with the functionality agreed to in advance, it is delivered on time and it is delivered within the agreed budget. When these three factors balance each other, we can speak of a successful project" [10].

It may be concluded that the definitions are not univocal. Some authors are of the opinion that a project fails when it does not achieve successful implementation, other authors take this further and include the success of the information system in the organization in relation with the user satisfaction and the benefits for the business in their assessment.

In order to be able to establish how the Dutch projects used in this paper agree or disagree with what others say happens, it is necessary to map out a number of publications and what others say happens in this field. To that purpose, the discussion examines the following publications:

International publications

- Large Software System Failures and Successes [12]
- Major Causes of Software Project Failures [13]
- Critical Success Factors In Software Projects [14]
- Seven Characteristics of Dysfunctional Software Projects [15]
- The 40 root causes of troubled IT projects [1]
- Critical failure factors in information system projects [2].

Dutch publications

- Why are information systems still failing? [5]
- Success and failure factors in complex ICT projects [9]
- ICT project management on the road to adulthood: Success factors for ICT projects [10].

The authors of these articles report on research carried out to determine the success and failure factors in ICT projects. Some authors have also published books on this subject (for example [16], [5], [12]). The procedures in Tarek Abdel-Hamid's book "Software Project Dynamics – An Integrated Approach" [4] are also examined for relevant procedural factors.

2.1 The project life-cycle and root causes of troubled projects

John Smith stated: "The focus of my book is troubled projects and I shan't be saying anything more about failed projects, other than to point out that all failed projects were once troubled projects. It is our goal to detect and correct the causes of a project's troubles before it fails.".
Some ICT projects are doomed before the ink is dry on the contract. Others fall prey to troubled project root causes which strike later in the project life-cycle. Every project passes through a

number of stages, as shown by John Smith in figure 1. The figure shows that most projects move through six stages. These stages often overlap to some extent. The stages are: Project Conception, Project Initiation/Mobilisation, System Design, System Development, System Implementation, and System Operation, Benefit Delivery, Stewardship and Disposal.

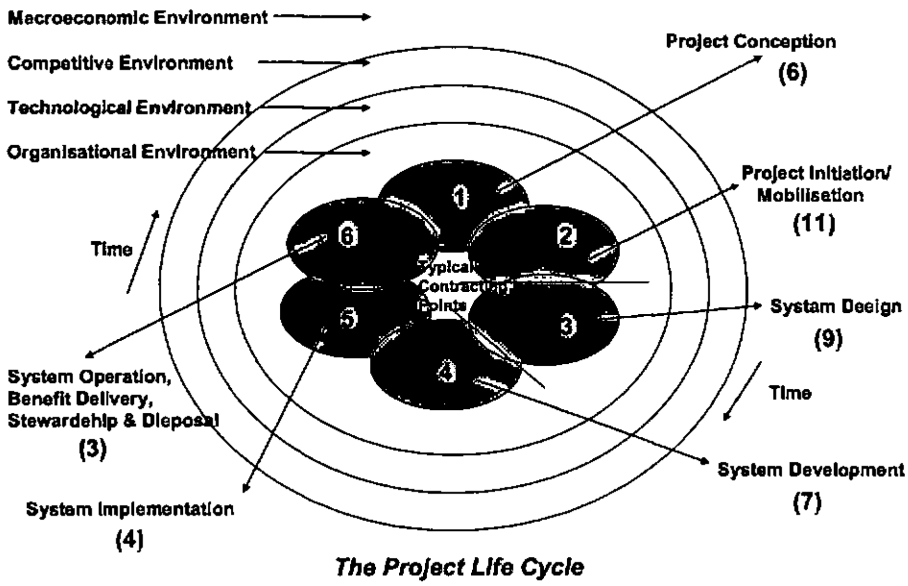


Figure 1: The Project Life Cycle [16]

ICT engagements, projects and programmes have a habit of getting into difficulty, for example the NUMIS-2000 project [17]. If the problems are acute, the project will fail and both buyer and vendor may suffer substantial loss, for example the WIA project [18]. More likely, it will become 'troubled', usually implying that it completes, but costs greatly exceed those budgeted and project time-scale is substantially longer than planned.
John Smith stated that: "The 'soft' and 'business' skills needed to successfully engage with clients and safely contract for services are seldom taught on graduate courses; these skills must be acquired 'on the job'. This takes time and organisations have limited success at passing on 'lessons learned'.".

It is an interesting fact that, after analysing several sources of information and using his own experience, John Smith was unable to find more than 40 generic root causes of troubled projects. His list is shown in table I.

John Smith distilled the problems into 40 generic root causes (RC) of troubled projects

Author	No	Description
Project Conception:		**JS/RC01 – JS/RC06 (6)**
JS	RC01	Project based on an unsound premise or an unrealistic business case
JS	RC02	Buyer failure to define clear project objectives, anticipated benefits and success criteria
JS	RC03	Project based on state-of-the-art and immature technology
JS	RC04	Lack of buyer board-level ownership/commitment or competence
JS	RC05	Buyer's funding and/or time-scale expectations unrealistically low
JS	RC06	Buyer failure to break a complex project into phases or smaller projects
Project Initiation/Mobilisation:		**JS/RC07 - JS/RC17 (11)**
System Design:		**JS/RC18 - JS/RC26 (9)**
System Development:		**JS/RC27 - JS/RC33 (7)**
System Implementation:		**JS/RC34 - JS/RC37 (4)**
System Oper., Benefit Delivery, Stewardship and Disposal		**JS/RC38 - JS/RC40 (3)**

Table 1: Root causes (RC) of troubled projects

Government ICT projects

Whilst it is true to say that government ICT projects are procured and managed rather differently from private sector projects, the report "Improving the delivery of government ICT projects" [19] has much valuable advice to offer. A number of "war story projects" are exposed. Summarising the key conclusion and recommendations of the report finds the following root causes of troubled projects in the UK public sector:

Author	No	Description
JS	PUBRC01	Lack of senior management involvement and commitment
JS	PUBRC02	Failure to focus on key business and end-user needs
JS	PUBRC03	Failure to break complex projects into manageable, separately contracted 'chunks'
JS	PUBRC04	Poor and unimaginative project management
JS	PUBRC05	Poor risk management and contingency planning
JS	PUBRC06	Unclear contracts and poor contract management
JS	PUBRC07	Insufficient focus on user training needs and the design of training interventions

Table 2: Root causes of troubled projects in the UK public sector

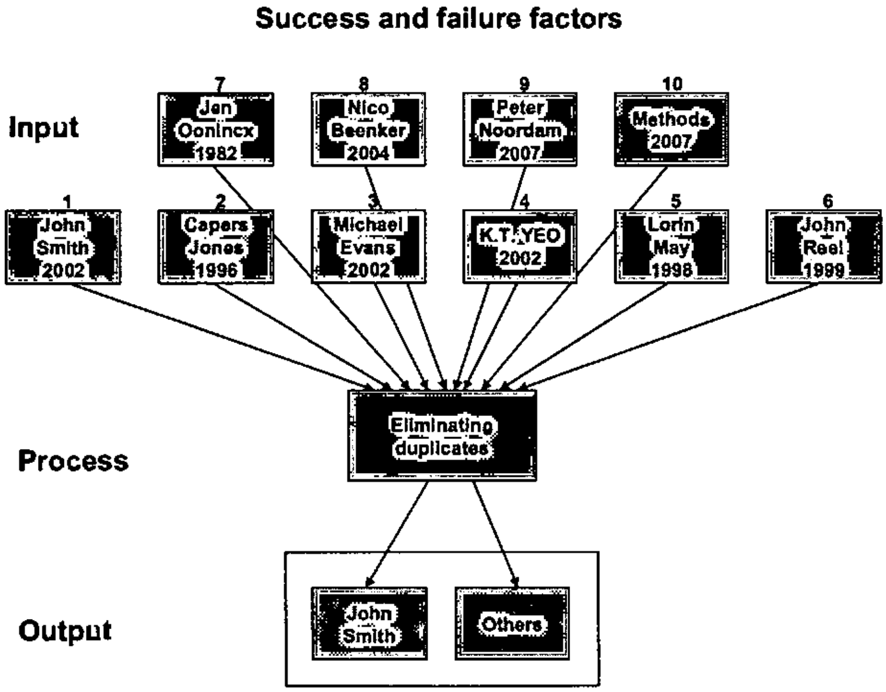


Figure 2: The "Eliminating duplicates" process

It is obvious that some success and failure factors are brought forward by several authors. By including these duplicates only once in the results, a collection of success and failure factors is created that may be used as a frame of reference. This is shown in figure 2. Together with "*the procedures in Tarek Abdel-Hamid's work on Software Project Management* (see section 2.4) and the "big hitters" (see section 2.6) the SUFFI-model (figure 3) has been created.
The SUFFI model consists of a number of tables and can be put to use immediately and further tested and improved. In order to simplify its use even more, the tables have been combined into one single table, the so-called reduced **SUFFI Chart** (figure 4). The SUFFI Chart consists of 139 SUFFIs and includes the research results of various researchers. Reduced means that duplicates and some other SUFFIs are eliminated. In order to be able to test new research results more easily against all SUFFIs, a **SUFFI Total Chart** was created in which all SUFFIs were included. Moreover, this SUFFI Total Chart consists of 247 SUFFIs and includes all known relations between the SUFFIs.

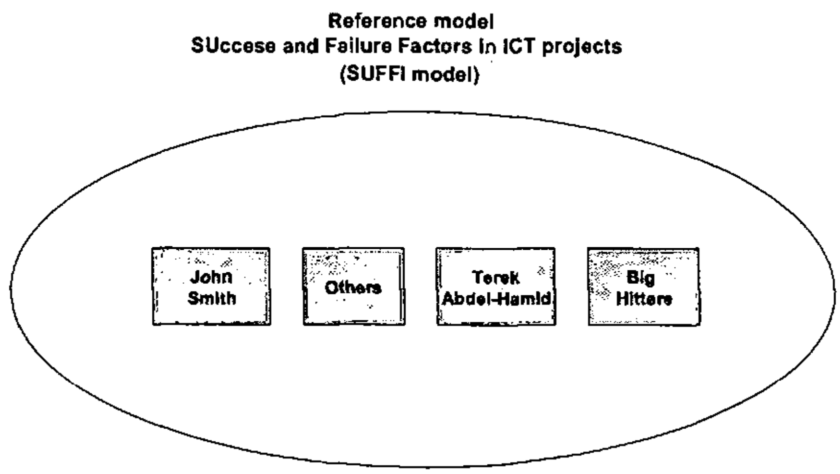


Figure 3: Reference model SUccess and Failure Factors in ICT projects (SUFFI model)

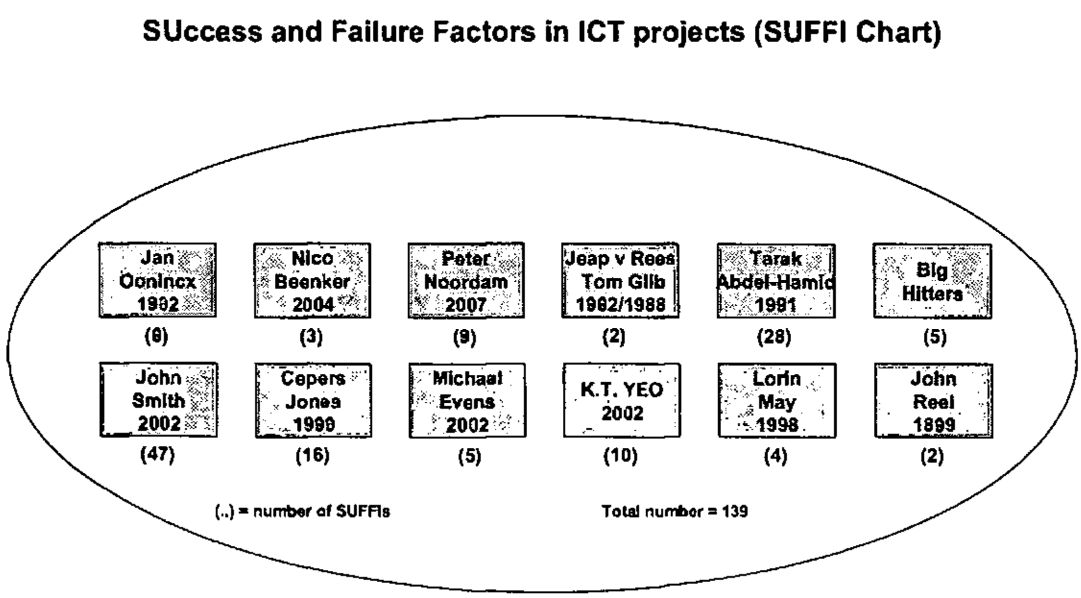


Figure 4: SUFFI Chart

2.2 Software project outcomes by size of project

An interesting question is whether the size of a project does play a part in success and failure factors, because a large project may result in a higher degree of complexity and a longer turn around time.

Failure factors mentioned by John Smith are for example: "Buyer failure to break a complex project into phases or smaller projects" (JS/RC06) en "Failure to break complex projects into manageable, separately contracted chunks" (JS/PUBRC03).

Professor Nielen [17] has the opinion that the realization-effort of a project relates quadratically to the size of the project. An information system of for example 6,000 function points is 9 times more difficult to realize than an information system of 2,000 function points because of growing complexity and increasing costs [17], [4].
Peter Noordam [10] says that: "We have looked at another important success factor: the project size. Looking at the project size in our study we see significant differences between banks and insurers on the one hand and the central government on the other hand. Very large projects worth more than 10 million Euros only take place within the central government; 64% of all projects at banks and insurers are worth less than 1.5 million Euros, whilst at the central government 43% of all projects are worth less than 1.5 million Euros. This could explain why the central government is less satisfied compared to banks and insurers. From the research carried out by the Standish Group International Inc [7] it can be concluded that it is sensible to set up projects such that these do not exceed the 1.5 million dollar line in order to increase the chance of success."
In 1996, Capers Jones [12] carried out research into "Large Software System Failures and Successes". In this research, he did not just look at the size of a project, but also studied the type of industry for which the software was developed. Capers Jones said final: "Computers and software have become indispensable in modem business, government, and military operations. The need for software has created one of the major occupations of the 20th century. However, software is a troubling technology. Software development is highly labour intensive, and as a result, large software projects are among the most expensive undertakings of the 20th century. For example, large software systems cost far more to build and take much longer to construct than the office buildings occupied by the companies that have commissioned the software".

2.2.1 Size of project

Capers Jones [12] states: "Really large software systems in the 100,000 function point (FP) size range can cost more than building a domed football stadium, a 50-story skyscraper, or a 70,000-ton cruise ship". Not only are large systems expensive, but they also have one of the highest failures rates of any manufactured object in human history. The term "failure" refers to projects that are cancelled without completion due to cost or schedule overruns or that run later than planned by more than 25 percent. He stated: "Let us consider what the phrase 'large systems' means in the context of six different size plateaus separated by an order of magnitude for each plateau: 1 FP (125 C statements); 10 FP; 100 FP; 1,000 FP; 10,000 FP; 100,000 FP (12,500,000 C statements)". He goes on to say: "Using these six size ranges, table 3 shows the approximate frequency of various kinds of outcomes, ranging from finishing early to total cancellation. Applications of 10,000 function points are often plagued by cost and schedule overruns and by outright cancellations. Applications that approach 100,000 function points in size are among the most troubling constructs of the 20th century". Based on this information it is wise to limit your project to 1,000 function points.

	Probability of Selected Outcomes				
	Early	**On time**	**Delayed**	**Cancelled**	**Sum**
1 FP	14.7%	83.2%	1.9%	0.3%	100%
10 FP	11.1%	81.3%	5.7%	2.0%	100%
100 FP	6.1%	74.8%	11.8%	7.3%	100%
1,000 FP	1.2%	60.8%	17.7%	20.3%	100%
10,000 FP	0.1%	28.0%	23.8%	48.0%	100%
100,000 FP	0.0%	13.7%	21.3%	65.0%	100%
Average	5.5%	56.9%	13.7%	23.8%	100%

Table 3: Software project outcomes by size of project

2.3 The present Dutch situation

From a scientific point of view, the Netherlands has enjoyed an excellent reputation with respect to software research [20] for many years. Occasionally, there were complaints about this research being very theoretically orientated. However, that is changing. The Ministries of Economic Affairs and Education Culture and Science and financial backer of research NWO decided in 2000 to give an impulse to software research with industrial relevance. Representatives from trade and industry and research institutes jointly set up a programme, which from 2002 was known as "JACQUARD" (Joint ACademic QUAlity Research and Development). JACQUARD resulted in an intensive interaction between researchers and a few dozens companies. By means of discussions on progress and user committees, the businesses ensure that research is embedded into everyday practice. An assessment committee consisting of renowned scientists and people from the business community assess the propositions for scientific quality, innovation and practical relevance for the business community.

In practice, it is still difficult to carry out ICT projects on time and within budget [5], [8], [9]. Peter Noordam [10] said: "We asked ourselves the question what the situation is in 2007, with regard to satisfaction on projects and whether the factors as mentioned are still blamed for the failure of projects. In doing so, we asked ourselves what we could learn from failing and successful ICT projects. We decided to carry out a project management study in order to obtain handles for successful project implementation, based on insight into the success and failure factors of ICT projects."

Study design

The study was carried out by means of a survey in the period between October and December 2006. The questionnaire was distributed amongst approximately 3000 IT professionals (managers, project managers and IT specialists) in the Netherlands and resulted in over 230 respondents (7.7%). The results of the survey were processed by two teams consisting of students of the Free University Amsterdam and tested against other research data. The interim results were analyzed with a selection of the interviewees. Next, during a group session in late December 2006 the results [21] were presented to and discussed with project managers from different branches.

Results

Peter Noordam said: "Because 'result' is difficult to assess – after all, result involves realization of the objectives (benefits) of a project as set in advance – for answering the survey we were mainly guided by the *degree of satisfaction with regard to projects.* Satisfaction is more qualitative and reflects a feeling with regard to the project. In the study, the respondents were asked to show their satisfaction with the projects in their organization by means of a report mark on a scale of between 1 and 10. This resulted in an average mark of 5.5 across the branches. Not a 'nice' average and still a lot of room for improvement". Table 4 contains some failure factors that were discussed by the study of Peter Noordam et al.

Author	**No**	**Description**
		The respondents named the following reasons as being the most important ones for the failure of projects:
PN	01	Poor project management
PN	02	Lack of communication in and around the project
PN	03	Objectives not defined
PN	04	Unfamiliarity with scope and complexity
PN	05	Technical complexity and technical integration issues
PN	06	The use of a business case results in a higher degree of satisfaction with the project, whilst the satisfaction with the project is very low when no business case is used
PN	07	The fact whether the costs are evaluated (during and after the project) and whether the benefits are evaluated (at the end of the project) also prove to have a fairly large influence on the success
PN	08	The satisfaction increases when a standard project management methodology is used
PN	10	Technical knowledge is certainly an important skill for project managers to have
PN	14	Evaluating projects is strongly related to the satisfaction about the successful execution of projects

Table 4: Some failure factors in development and introduction of information systems

Methods for developing information systems (and software) have been receiving attention for a long time. In the Netherlands during 1981 and 1982, a series of no less than 24 articles appeared in the magazine Informatie. One of these articles is "De methode doet het niet" (The method does not do it) by Jaap van Rees [22]. Amongst other things, he draws the conclusion that: *"It is the designer who designs, not the method"*. Tom Gilb [23] also takes up a critical position towards the method: *"Don't believe blindly in any one method; use your methods and common sense to measure the reality against your needs"*.

2.4 Success/failure factors of "Software Project Dynamics – An Integrated Approach"

For each chapter of this book the main subjects of *"the procedures in Tarek Abdel-Hamid's work on Software Project Management"* have been mapped point by point. Based on this information of the 20 chapters, 82 SUFFIs are derived. Some of these are shown in table 5. The 82 SUFFIs are subdivided into two categories, namely: category 1 and category 2. The SUFFIs in category 1 are, in my opinion, the main SUFFIs and these are therefore included in the SUFFI Chart. The SUFFIs in category 2 are only included in the SUFFI Total Chart.

Author	No	Description
TAH	06	Brooks' Law: adding more people to a late software project makes it later
TAH	09	System complexity grows as the square of the number of system elements; therefore, experience with a small system cannot account for all the things that will have to be done in a large system. Neither will the Experience Method apply to systems of totally different content
TAH	11	It is difficult to measure performance in programming
TAH	14	The relationship between cost and system size is not linear. In fact, cost increases approximately exponentially as size increases
TAH	24	The project goals and schedules can play a significant motivational role throughout the life of a software development project
TAH	34	Progress, especially in the earlier phases of software development, is measured by the rate of expenditure of resources rather than by some count of accomplishments
TAH	52	By imposing different estimates on a software project we create different projects. Research clearly indicates that the decisions that people make in projects and the actions they take are significantly influenced by the pressures and perceptions produced by the project's schedule
TAH	54	When the project is perceived to be ahead of schedule, "Parkinson's Law indicates that people will use the extra time for ... personal activities, catching up on the mail, etc.". This, of course, means that they become less productive
TAH	57	The "Safety Factor Policy" does achieve its intended objective: more accurate estimates. However, the organization pays dearly for this
TAH	64	Our model results indicate "adding manpower to a late software project makes it more costly, but *not necessarily later*"
TAH	65	A different distribution of estimated effort among a project's phases creates a different project

Table 5: Some success/failure factors based on the procedures of Tarek Abdel-Hamid (TAH)

2.5 Summary

- some TAH SUFFIs are **additional** to the SUFFIs/Root Causes from John Smith and the SUFFIs from others, for example: TAH/09, TAH/14, TAH/52, TAH/64 and TAH/65;
- not all TAH SUFFIs are "new SUFFIs" but they mainly **provide more insight**. "Poor Management" is for example often mentioned by authors. However behind "Poor

Management" a whole range of subjects are hidden that may contribute to this SUFFI.

2.6 Which root causes are the 'big hitters'?

John Smith says: "It is *unwise* to try to rank the root causes into 'big hitters' and 'the rest'. However, such ranking is definitely of value for an *individual* project as it will inform the prioritisation of 'turnaround' actions". With regard to this particular point, John Smith disagrees with other researchers (Evans, Yeo, Reel, Beenker, Noordam) that do indicate a ranking of success/failure factors. Table 6 includes the success/failure factors that have been named as "big hitter" by at least 4 authors.

Together the tables form the ***"Reference model success and failure factors in ICT projects"*** *(for short:* ***SUFFI model****, SUFFI = SUccess and Failure Factors in ICT projects) (figure 3).* That establishes: "The opinion of others about Software Project Management".

Success/failure factors that are mentioned the "big hitters" (BH)	**CJ**	**ME**	**KY**	**LM**	**JR**	**JO**	**NB**	**PN**	**Total**
Poor project management (BH01)	+	+	+	+	+	+	+	+	8
Deadlines are unrealistic (BH02)		+	+		+		+		4
Poor communication (BH03)			+	+			+	+	4
Incomplete/weak definition requirements (BH04)			+	+	+			+	4
Insufficient involvement of future users (BH05)			+	+	+	+			4
John Smith: "It is *unwise* to try to rank the root causes into 'big hitters' and 'the rest'. However, such ranking is definitely of value for an *individual* project as it will inform the prioritisation of 'turnaround' actions."									
Poor project management and poor communication are mentioned by Tarek Abdel-Hamid, but not as a "big hitter". Requirements definition phase and acceptance testing phase are excluded because they both lie outside the boundary of the TAH-model (responsibility of the user organization)									

Table 6: Success/failure factors that constitute the "big hitters"

3 THE PORTFOLIO OF PROJECTS

3.1 Introduction

The nine projects represent an effort and duration of about sixteen years. The first author's role in projects 6-9 was: *internal* project manager at the Delft University of Technology and the projects 1-5, the *external* project manager. In his role as a certified IT auditor he audited four ICT projects. These four audits were added to the portfolio, so the portfolio consists of nine projects and four audits (table 9) [24]. In this section one of the projects will come up for extensive discussion and all projects and audits are tested against the "big hitters".

3.2 The Telephony project

In 1993, the Delft University of Technology (DUT) carried out the project "Vervanging Telefooncentrale" (Replacement Telephone Exchange). The objective of the project was to bring a new telephone exchange (PABX) with peripheral equipment into production, which would comply with the functional requirements and to offer Telephony Services (TS) as a service of the Computer Centre. Replacement of the telephone exchange also meant that all the small telephone exchanges on the campus were made defunct and resulted in rollout of new telephones for a large part of the approximately 7,000 connections. Extension of the cabling on the campus and its buildings was also planned.

Organization
For the benefit of the 'introduction' phase, a project organization was formed that consisted of a steering committee and a project group. The steering committee was chaired by the managing director of the Computer Centre (DIR-RC). The head Operations (HOPS) of the Computer Centre and a representative of the suppliers (DIR-SUPPL) were also part of the steering committee. The project group consisted of the general project manager (GPM), an external telecom consultant (ExCons) and four working groups, namely:

- Marketing, Information & Training (MIT), under supervision of PM-MIT;
- Inventory & Implementation (I&I), under leadership of PM-I&I;
- Functions (FU), under the leadership of PM-FU;
- Management & Operation (M&O), under the leadership of PM-M&O.

The role of the first author was that of external project manager (PM-M&O).

Table 7 contains the applied SUFFIs from the SUFFI Chart. Table 8 contains the result of the Telephony project.

Author	**No**	**Description**	**Apply to**
		Big Hitters	
BH	01	Poor project management	No
BH	02	Deadlines are unrealistic	No
BH	03	Poor communication	No
		Remarks/Observations: The MIT working group ensured excellent communication with the entire university.	
BH	04	Incomplete/weak definition requirements	Yes
BH	05	Insufficient involvement of future users	No
		JS+Others	
JS	PUBRC01	Lack of senior management involvement and commitment	No

		Remarks/Observations: Involvement and commitment were good. However, it should be said that blind faith of a manager in one single employee without remaining critical is very hazardous and, as happened in this case, may lead to a "troubled project".	
JS	RC05	Buyer's funding and/or time-scale expectations unrealistically low	Yes
		Remarks/Observations: DIR-RC: "In my entire career, this is the most underestimated project 1 have ever been involved in".	
JS	RC09	Failure to achieve an open, robust and equitable buyer-vendor relationship	No
		Remarks/Observations: Cooperation between project group and supplier was excellent!	
KY	08	Poor internal communication	Yes
		Remarks/Observations: Within the project team, PM-I&l caused bad communication, which resulted in a "troubled project".	
NB	05	Problems escalated too late	Yes
PN	10	Technical knowledge is certainly an important skill for project managers to have	Yes
		Remarks/Observations: In view of the technical complexity of the project, the project manager required technical knowledge. Because there was little knowledge of telephony within the Computer Centre, the ExCons was hired. He graduated from the DUT and was very familiar with the environment.	
		Tarek Abdel-Hamid*)	
TAH	02	Poor management can increase software costs more rapidly than any other factor	Yes
TAH	08	Two factors significantly influence the initial estimate of the job size: (1) the firm's previous experience; and (2) the general over-all tendency to underestimate the job size	Yes
TAH	09	System complexity grows as the square of the number of system elements; therefore, experience with a small system cannot account for all the things that will have to be done in a large system. Neither will the Experience Method apply to systems of totally different content	Yes
TAH	10	In a multi-project environment, competition for company resources becomes a significant dimension	Yes
TAH	12	The longer an error goes undetected, the more extensive the necessary rework and the greater the cost	Yes

TAH	28	Design errors in the early design phase are generated at a higher rate than are coding errors	Yes
TAH	36	There is a powerful tendency to focus on the highly visible mainline components of the software, and to underestimate or completely miss the unobtrusive components (e.g., help message processing, error processing, and moving data around)	Yes
TAH	37	In some cases the management becomes increasingly willing to "pay any price" necessary to avoid overshooting the "Maximum Tolerable Completion Date". In such cases, management is often willing to hire more people	Yes
TAH	42	Using the system dynamics model as an experimentation vehicle can provide many benefits to the software engineering community	Yes
TAH	44	By using a model of a complex system, more can be learned about internal interactions than would ever be possible through manipulation of the real system	Yes
TAH	50	For many years, estimation of software project development time and cost had been an intuitive process based on experience and analogy	Yes
TAH	64	Our model results indicate "adding manpower to a late software project makes it more costly, but *not necessarily later*"	Yes
TAH	65	A different distribution of estimated effort among a project's phases creates a different project	Yes
TAH	81	An important implication is that a software estimation model cannot be judged solely based on how accurately it estimates historical projects	Yes
		*) The list is not exhaustive	

Table 7: Success/failure factors Telephony project

Telephony	**Score**
Complies with functionality agreed	Yes
On time	Yes
Within the agreed budget	No

Table 8: Results Telephony project

Remarks

The SUFFIs provide more detailed understanding. If the management had been aware of the SUFFIs mentioned in table 7 then the project would have probably run more smoothly and the costs would have been lower. Furthermore, in that case the following observations probably would not have been applicable:

- involvement and commitment were good. However, it should be said that blind faith of one manager in one particular employee without being sufficiently critical is very hazardous and, as happened in this case, may lead to a "troubled project";
- the managing director of the Computer Centre stated: "In my entire career, this is the most underestimated project I have ever been involved in";
- within the project team, the manager of Inventory & Implementation caused poor communication, which resulted in a "troubled project".

3.3 The portfolio projects and the Big Hitters

The SUFFI Chart includes the SUFFIs that have been named as Big Hitters by at least 4 authors. All thirteen cases (9 projects and 4 project audits) were tested against the Big Hitters.
Although the results in table 9 are based on a very limited spot check, the conclusion may be drawn that the Big Hitters within this collection of cases indicate the failures quite readily. Where at least four of the five Big Hitters are not applicable, the "score" is positive. Where at least four of the five Big Hitters are applicable, the "score" is negative. This picture also corresponds with my experiences and observations during other projects and audits. When the five Big Hitters lead to a negative score, a large number of other SUFFIs usually play a part. It is important to note the poor results for the audits where the author was called in because of these difficulties. Even with no major identified problem features many of the projects still had problems.

	Apply to					Performance		
Results cases **)	Big Hit-ter 1	Big Hit-ter 2	Big Hit-ter 3	Big Hit-ter 4	Big Hit-ter 5	Funct.	On time	Within Budget
Case 1: POTVIS project (KLPD)	No	No	No	No	No	Yes	Yes	Yes
Case 2: Kolibrie project (KPN Telecom)	No	No	No	No	No	Yes	Yes	Yes
Case 3: Charging method project (GAK)	No	No	No	No	No	Yes	Yes	Yes
Case 4: Telephony project (DUT)	**No**	**No**	**No**	**Yes**	**No**	**Yes**	**Yes**	**No**
Case 5: OKAPI project (UoA)	No	No	No	No	No	Yes	Yes	---#
Case 6: GIRAF project (DUT)	No	No	No	No	No	Yes	Yes	---#
Case 7: AUBID project (DUT)	No	No	No	No	No	Yes	Yes	---#
Case 8: VDV project (DUT)	No	No	No	No	No	Yes	Yes	---#
Case 9: BIBLIOSYSTEM project (DUT)	No	No	No	No	No	Yes	Yes	---#
--	------	------	------	------	------	------	-----	-------
Case 10: Audit Multihouse	Yes	Yes	Yes	Yes	---+	No	No	No
Case 11: Audit SYSA (GOVERN)	Yes	Yes	Yes	Yes	Yes	No	No	No
Case 12: Audit ACCINT (PUBLIC)	Yes	Yes	Yes	Yes	Yes	No	No	No
Case 13: Audit SOX (FINANCE)	No	No	Yes	No	No	Yes	Y/N*	Y/N*

+) unknown
#) no specific budget available
*) Yes or No, depends on the project
**) table 10 gives some details of all cases (projects and audits)

Table 9: Failure Big Hitters in relation to the case studies

4 OUTLINE OF OTHER PROJECTS

Section 3 contains an extended description of the Telephony project (Delft University of Technology) and table 9: the Big Hitters in relation to all the cases.
In this section table 10 gives some details of all cases (projects and audits) [24]:
name of the case, company, remarks and for the projects: staff and size software. These cases are outlined in relation to the SUFFIs in table 9.

No	Project with external project-based publications	Company / Remarks	Period	Duration in years
1	POTVIS (PrOjecT Verbetering Infrastructuur SAP) (Project improvement infrastructure SAP)	KLPD (National Police Services Agency) • *Improvement to infrastructure in a complete SAP-environment: network segregation, RBAC (Role-based Access Control), backup and restore, etc. (109 improvements).* • *Presentation of the results of RBAC at KLPD for the Dutch Society for Information Science NGI (80 participants).* **Staff** The project manager was dedicated to the project. Among other things he designed the RBAC model. About fifteen staff members were part-time available for doing different jobs. **Size software** No application software development.	2003-2004	1.7
2	Kolibrie (Kpn On Line Interfacing met Baan, Resultaat Is Effectief) (Interfacing with BaanERP)	KPN (Dutch Telecom company) • *Interfacing BAAN-ERP with other information systems (based on EAI (Enterprise Application Integration): message broker, adapters).* • *Presentation of the results of EAI at KPN for the Dutch Society for Information Architects (90 participants).* **Staff** The project manager was dedicated to the sub project Interfacing. Among other things he designed the EAI architecture together with a BAAN-ERP interfacing specialist. Four other people participated in the sub project Interfaces. **Size software** 20,000 lines of code (estimated).	1999-2001	2

3	GAK (Charging method for the use with National databases)	GAK/ASZ (Gemeenschappelijk Administratie Kantoor/Automatisering Sociale Zekerheid) • *Development of a charging method based on functional services (e.g. functional transaction codes).* • *Developed for the use with National databases (containing employees, employers and their contractual obligations).* **Staff** The project manager/research engineer was dedicated to the research project. He developed the new charging method. About ten people gave input to the project. **Size software** No application software development.	1997-1998	0.5
4	IMPALA (Information Management PAbx and Local Administration)	DUT (Delft University of Technology) • *Implementation of a new PABX (Private Automatic Branch eXchange) with 7,000 telephone connections.* • *Charging modules, management procedures, information management, etc.* • *Project-based thesis Executive Master of IT Auditing (EMITA) EUR.* **Staff** The project manager was dedicated to the sub project. Among other things he designed the IMPALA model. Two other staff members were available for doing different jobs. **Size software** 10,000 lines of code (estimated).	1993-1994	2

5	OKAPI (Online KAPItaaldienst) (Online capital services)	UoA (University of Amsterdam) • *Development of a new decentralized information system concerned with the financial building administration.* • *Including two-phase RBAC.* **Staff** The project manager was dedicated to the project. Among other things he designed the functional design. Four other staff members were available for doing different jobs. **Size software** 90,000 lines of code (estimated).	1991-1994	2
6	GIRAF (General Information RetrievAl Facilities)	DUT • *General Information Retrieval Facilities on mainframe computers.* • *Many applications on the DUT.* • *Flexible multi-language information system.* • *Adaptations possible without changing the system.* • *Sold to other Universities and companies in the Netherlands and Belgium.* • *After the project-based publication, some companies in the Netherlands & Belgium adapted the architectural/engineering solutions for all applications in their company.* **Staff** The project manager was dedicated to the project. Among other things he designed the functional design and architecture of GIRAF. About three staff members were available for doing different jobs. **Size software** 160,000 lines of code.	1982-1984	2.5

7	AUBID (AUtomatisering BIbliotheek DUT) (Computerization of the Library DUT)	DUT • *Information system related to book reservations and book requests.* • *Based on CICS/VS (Customer Information Control System/Virtual Storage).* • *Some solutions, published in the project-based publication, are implemented by dozens companies (especially the solution of the printer problem).* **Staff** The project manager was dedicated to the project. Among other things he designed the functional design. Four staff members were part-time available for doing different jobs. **Size software** 20,000 lines of code (estimated).	1977-1980	2.5
8	VDV (Verkeers Data Verzameling) (Traffic Data Collection on behalf of traffic research engineers)	DUT • *Unique system to collect traffic data on behalf of the Research of Traffic Streams.* • *Very difficult to develop (hardware, software) because it was not clear in advance if a solution to the problem of the users (traffic engineers) could be found.* • *Engineering system for the traffic engineers of the DUT.* • *Has been used for more than 25 years.* **Staff** The project manager was dedicated to the project. He designed and programmed the VDV model. Two staff members were part-time available for doing different jobs. **Size software** 6,000 lines of code.	1972-1974	3

9	BIBLIOSYSTEM (Batch oriented information retrieval system for general purpose)	DUT • *Batch oriented information retrieval system.* • *Many applications, including bibliographies.* • *After the project-based publication, more than 100 companies asked for the documentation/sources.* **Staff** The project manager was dedicated to the project. He designed and programmed the BIBLIOSYSTEM information system. One staff member was part-time available for doing different jobs. **Size software** 2,000 lines of code.	1970-1971	0.5
		Subtotal		***16.7***
No	**IT-audits**	**Company / Remarks**	**Period**	**Duration in years**
10	Multihouse (New business information system for public utilities companies)	Multihouse BV • *Information system NUMIS-2000 developed by Multihouse BV.* • *Ordered by SWV NUMIS-2000.* • *Audit ordered by Court of Justice in Amsterdam.* • *Audit team (Commission of Third Party Experts): Prof.dr.ir. M. Looijen, Prof.dr.ir. G. C. Nielen and ir. A. J. van Dijk EMITA RE.* • *Both of the parties did not realize the degree of complexity.*	1997	0.2

11	SYSA (Registration of report data and maintenance of equipment)	GOVERN • *Replacement of PROGA and PROGB by SAP R/3.* • *Although a lot of money had been invested in SAP/SYSA already, it was recommended to stop the project in order to avoid further escalation.* • *The head of ICT-Operations of the Information Provision Service had proposed to have a brief investigation (audit) done by an independent researcher/auditor.* • *The first author was asked to carry out this investigation. He accepted the commission under the express condition that he would be able to carry out an independent/impartial investigation.* • *It is interesting that six months on my advice with regard to PROGA was taken on board fully and an investigation into solution approach 1 with regard to PROGB as recommended by me was underway.*	2005	0.1
12	ACCINT (Access to Internet)	PUBLIC • *One of the objectives of the ACCINT project was: introduction of the Internet functionality (both the e-mail as well as the browse functionality) in a controlled and as safely as possible way for obvious users.* • *The "Rollout of the browse functionality" of the ACCINT project did not go entirely as planned.* • *The CIO (Chief Information Officer) of PUBLIC had commissioned the first author to carry out an audit/ investigation into ACCINT with the objective to provide an answer to the question "What went wrong?".* • *Although ICT-Operations had been receiving the necessary criticism in the report, the head of the department did agree with the conclusions.*	2005	0.1

13	SOX (Testing the Sarbanes-Oxley Act)	FINANCE • *As part of SOX (the Sarbanes-Oxley Act), FINANCE, a well-known financial institution in the Netherlands, carried out audits/tests on the IT infrastructure.* • *The first author was one of the IT auditors and carried out audits on a number of realized IT projects.* • *SOX means: the adage "tell me" has changed into "show me". This has a major impact on the companies themselves, which was also obvious during these project audits.*	2006	0.4
		Subtotal		***0.8***
		Total		***17.5***

Table 10: Portfolio of projects

5 PROJECTS RESULTS

Table 9 contains the conclusions of all cases. Section 3 contains the results of the Telephony project. This project was tested against the Big Hitters from the SUFFI Chart (table 7). Based on the results of the tests the performance was evaluated. In the same way the performance of the other twelve cases from the portfolio were concluded and presented in table 9 [24]. Many of the issues outlined by other authors were present and in general a number of important conclusions can be drawn.

6 CONCLUSIONS

The above mentioned study shows that the subject of project success and failure is still very relevant in the year 2008. It may be concluded that the subject of success and failure factors in ICT projects has been in the spotlight for more than 26 years and still is very topical.
A chart outlining 139 SUFFIs (SUccess and Failure Factors in ICT projects) has been devised to give guidance to project managers. Some TAH (Tarek Abdel-Hamid) SUFFIs are additional to the SUFFIs in the SUFFI Chart and are only included in the SUFFI Total Chart. Not all of these are "new SUFFIs" but they do provide more insight. "Poor Management" is for example often mentioned by authors. However behind "Poor Management" a whole range of problems are hidden that may contribute to this SUFFI.

The SUFFI Chart includes the SUFFIs that have been named as a Big Hitter by at least 4 authors. All thirteen cases (9 projects and 4 project audits) were tested against the failure Big Hitters. Although the results in table 9 are based on a very limited spot check, the conclusion may be drawn that the collection Big Hitters within this portfolio of cases acts discriminating. Where at least four of the five Big Hitters are not applicable, the "score" is positive. Where at least four of the five Big Hitters are applicable, the "score" is negative. This picture also corresponds with my experiences and observations during other projects and audits. When the five Big Hitters lead to a negative score, a large number of other SUFFIs also play a part.

Based on the information of Capers Jones in table 3 it is wise to limit your projects to 1,000 function points. The projects in this portfolio were limited to about 1,000 function points and were successful. So the results of the projects in this portfolio can't refute the conclusion based on the information in table 3. Unfortunately the size of the audited projects is unknown.

The SUFFIs provide more understanding. If the management of the cases 4, 10, 11 and 12 had been aware of the SUFFIs mentioned in the SUFFI Chart then the project would have been run more smoothly and the costs would have been lower.

Both experienced as well as inexperienced project managers can reap the immediate benefits. Spending a few hours in advance on studying the mapped SUFFIs in the SUFFI Chart will help them avoid a number of pitfalls.

Further research
The results of the research are recorded in the SUFFI Chart above. For reasons of major social interest, the subject of success/failure factors will demand the necessary attention over the next few years. Both researchers as well as project managers are invited to test the SUFFI Chart against their own research results and practical experience. This will enable further improvement

of the SUFFI Chart, serving all those involved in ICT projects well.

We propose a thesaurus of SUFFIs (similar to the ISO 9126-standard quality model).

References

1. Smith J., The 40 root causes of troubled IT projects, *Computing and Control J*, 2002, June: pp. 109-112
2. Yeo K.T., Critical failure factors in information system projects, *Int. J. Project Management*, 2002, 20: pp. 241-246
3. Royal Academy of Engineering, *The challenges of Complex IT projects*, Report of working group of RAE and BCS, 2004
4. Abdel-Hamid, Tarek and Stuart E. Madnick, *Software Project Dynamics: An Integrated Approach*, 1991, Englewood Cliffs, NJ: Prentice Hall
5. Oonincx J.A.M, *Waarom falen informatiesystemen nog steeds?*, Samsom, Alphen a/d Rijn, 1982
6. Berghout, Egon, *Informatietechnologie, een bodemloze put?*, Intreerede Rijksuniversiteit Groningen, 2002, 22 oktober
7. The Standish Group International Inc., *Standard Group CHAOS Report*, 1998
8. The Standish Group International Inc., *Standard Group CHAOS Report*, 25 March 2003
9. Beenker, Nico, *Studie naar succes- en faalfactoren van complexe ICT projecten*, Vrije Universiteit Amsterdam, 2004, May
10. Noordam, Peter, Norman Martijnse and Berry Derksen, ICT-projectmanagement op weg naar volwassenheid, *Informatie*, 2007, March: pp. 20-26
11. Netherlands Court of Audit, *Lessen uit ICT-projecten bij de overheid Deel A*, Den Haag, 2007, November
12. Capers Jones, *Large Software System Failures and Successes*, American Programmer, 1996, April: pp. 3-9
13. May, Lorin J., Major Causes of Software Project Failures, *CrossTalk*, 1998, July: pp. 9-12
14. Reel, John S., Critical Success Factors In Software Projects, *IEEE Software*, 1999, May/June: pp. 18-23
15. Evans, Michael W., Alex M. Abela and Thomas Beltz, Seven Characteristics of Dysfunctional Software Projects, *CrossTalk*, 2002, April: pp. 16-20
16. Smith J., *Troubled IT projects: prevention and turnaround*, 2001, IEE
17. Dijk, Aart van, Maarten Looijen and Gert Nielen, *Onderzoek NUMIS-2000 ten behoeve van de Arrondissementsrechtbank te Utrecht*, Zoetermeer, 1997, February
18. Toet, D., Managers en budget nekken WIA-project, *Computable*, 2008, 18 July: pp. 4
19. House of Commons, *First Report: Improving the delivery of government IT projects*, Session, Committee of Public Accounts, 1999-2000
20. Aarts et al., Emile, *Software Engineering*, Joint ACademic QUAlity Research and Development (JACQUARD), NWO, The Hague, 2006/2007
21. Business & IT Trend Institute, *Projectmanagementenquête 2006/2007. Handvatten voor succesvolle projecten*, Vrije Universiteit Amsterdam / BISNEZ Management, Bergambacht, 2007
22. Rees, Jaap R. van, De methode doet het niet, *Informatie*, 1982, February: pp. 81-93

23. Gilb, Tom, *Principles of Software Engineering Management*, Addison-Wesley Publishing Company, 1988
24. Dijk, Aart J. van, *Success and failure factors in ICT projects: A Dutch perspective,* presentation at Middlesex University, School of Engineering and Information Sciences, London, 2008

Abbreviations

ACCINT	ACCess INTernet
ASZ	Automatisering Sociale Zekerheid
AUBID	AUtomatisering BIbliotheek DUT
AvD	Aart van Dijk
BH	Big Hitter
CICS/VS	Customer Information Control System/Virtual Storage
CIO	Chief Information Officer
CJ	Capers Jones
DIR	Director
DUT	Delft University of Technology
EAI	Enterprise Application Integration
EMITA	Executive Master of IT Auditing
ERP	Enterprise Resource Planning
EUR	Erasmus University Rotterdam
ExCons	External Consultant
FP	Function Point
FU	Functions
GAK	Gemeenschappelijk Administratie Kantoor
GIRAF	General Information RetrievAl Facilities
GPM	General Project Manager
HOPS	Head Operations
I&I	Inventory & Implementation
IMPALA	Information Management PAbx and Local Administration
ICT	Information and Communication Technology
IR	Ingenieur
ISO	International Standard Organization
IT	Information Technology
JACQUARD	Joint ACademic QUAlity Research and Development
JO	Jan Oonincx
JR	John S. Reel
JS	John Smith
KLPD	Korps Landelijke Politie Diensten
KPN	Koninklijke Ptt Nederland
KOLIBRIE	Kpn On Line Interfacing met Baan, Resultaat Is Effectief
KY	K.T. Yeo
LM	Lorin May
M&O	Management & Operation
ME	Michael Evans
MIT	Marketing, Information & Training

NB	Nico Beenker
NGI	Nederlands Genootschap voor Informatica
NIST	National Institute of Standards and Technology
NWO	Nederlandse Organisatie voor Wetenschappelijk Onderzoek
OKAPI	Online KAPItaaldienst
PABX	Private Automatic Branch eXchange
PM	Project Manager
PN	Peter Noordam
POTVIS	PrOjecT Verbetering Infrastructuur SAP
PUBRC	Root cause in the UK public sector
RBAC	Role-Based Access Control
RC	Root cause
RC	Rekencentrum
SAP	Systems, Applications and Products in Data Processing
SD	System Dynamics
SOX	Sarbanes-Oxley Act
SUFFI	SUccess and Failure Factors in ICT projects
SUPPL	Supplier
SYSA	System A
TAH	Tarek Abdel-Hamid
TS	Telephony Services
UK	United Kingdom
UoA	University of Amsterdam
VDV	Verkeers Data Verzameling

Publication 2

Succes- en faalfactoren bij ICT-projecten

Bekend maar onbemind

Success and failure factors in ICT projects
Well-known but unpopular

2008

Don't believe blindly in any one method;
use your methods and common sense to
measure the reality against your needs.

Tom Gilb [1988]

This article previously appeared in:

Informatie, July/August 2008, pages 40-44

Abstract
The success and failure factors in ICT projects (SUFFIs) that have been formulated as a result of a report of the Netherlands Court of Audit are not new but have been largely known for some years. Nevertheless, (project) managers in ICT projects usually give little attention to SUFFIs in advance. If (project) managers would study the SUFFI Chart at the start of an ICT project, then they would be able to avoid a number of pitfalls. In this article, the findings of the Netherlands Court of Audit and the reactions of experts are tested by means of a SUFFI Chart.

succes/faal-
factoren

Succes- en faalfactoren bij ICT-projecten

Bekend maar onbemind

Volgens een rapport van de Algemene Rekenkamer is het slecht gesteld met de aanpak van ICT-projecten bij de overheid. Naar aanleiding van het rapport hebben experts diverse succes- en faalfactoren bij ICT-projecten genoemd. De auteur toetst de bevindingen aan de hand van een SUFFI-Chart.

Aart van Dijk

Eind november 2007 heeft de Algemene Rekenkamer van Nederland het rapport *Lessen uit ICT-projecten bij de overheid – Deel A* uitgebracht. De Algemene Rekenkamer komt hierin tot de conclusie dat het slecht gesteld is met de aanpak van ICT-projecten bij de overheid. De Nederlandse overheid geeft jaarlijks miljarden uit aan ICT-projecten die geheel of gedeeltelijk mislukken. Het onderzoek werd uitgevoerd naar aanleiding van Kamervragen eerder in 2007. Diverse experts hebben hun commentaar op het rapport gegeven en hebben een of meer succes- en faalfactoren bij ICT-projecten genoemd. De vraag kan worden gesteld of de aangedragen opmerkingen en adviezen nieuw zijn of dat we kunnen spreken van een hernieuwde belangstelling. Daarom heb ik in mijn proefschrift over ICT-succes- en faalfactoren in relatie tot mijn portfolio van ICT-projecten van de afgelopen veertig jaar, een hoofdstuk over deze casus van de Algemene Rekenkamer opgenomen. Daarnaast heb ik een zogenoemde SUFFI-Chart geconstrueerd (SUFFI staat voor SUccess-/Failure Factors in ICT projects) (Van Dijk, 2008). In dit artikel ga ik eerst in op de vraag of succes- en faalfactoren bij ICT-projecten nieuw zijn. Daarna toets ik de resultaten van een analyse van het rapport van de Algemene Rekenkamer en de aangedragen opmerkingen en adviezen aan de SUFFI-Chart.

Relevantie

De vraag kan worden gesteld of het relevant is om te kijken naar succes- en faalfactoren bij ICT-projecten en of deze nieuw zijn. Het aantal publicaties over dit onderwerp geeft aan dat veel onderzoekers zich bezighouden of hebben beziggehouden met dit onderwerp. Al in 1982 schreef professor Jan Oonincx het boek *Waarom falen informatiesystemen nog steeds?* Op uitnodiging van PRISMA – Lectuurvoorlichting te Voorburg heb ik in 1982 over dit boek een recensie geschreven:

Hoewel er veel goed werkende informatiesystemen bestaan, is het aantal informatiesystemen dat slecht functioneert of dat tijdens de bouw het oorspronkelijke einddoel, de ingebruikname, niet bereikt, legio. Dit boek behandelt, in zeventien hoofdstukken gegroepeerd, een aantal aspecten dat het ontwikkelen en invoeren van informatiesystemen zo vaak doet mislukken. Informatiesystemen die te ambitieus, te geïsoleerd of zonder goede planning worden opgezet, hebben een zeer grote kans te mislukken. Ook het onvoldoende betrekken van de toekomstige gebruikers bij de ontwikkeling van een informatiesysteem of een passieve houding van de topleiding leiden vaak tot teleurstellende resultaten. Het boek is geschikt voor studenten informatica, bedrijfskunde, economie etc. Hoewel de informaticus die in de praktijk werkzaam is, een aantal

Samenvatting

De succes- en faalfactoren bij ICT-projecten (SUFFI's) die zijn geformuleerd naar aanleiding van een rapport van de Algemene Rekenkamer zijn niet nieuw maar voor een groot deel al jaren bekend. Toch besteden (project)managers bij ICT-projecten vooraf doorgaans weinig aandacht aan SUFFI's. Als (project)managers aan het begin van een ICT-project de SUFFI-Chart zouden bestuderen, zouden ze een aantal valkuilen kunnen voorkomen.

bekende zaken zal tegenkomen, is het ook voor hem nuttig het boek aandachtig te lezen. (Toekomstige) gebruikers van informatiesystemen zullen veel baat hebben bij bestudering van dit boek. Het argument 'geen tijd' gaat hier niet op.

Het is een prettig leesbaar boek van bescheiden omvang maar met een zeer belangrijke inhoud.

In zijn intreerede aan de Rijksuniversiteit Groningen in 2002 zegt professor Egon Berghout

nr.	succes- en faalfactoren	gerelateerde SUFFI's
AR1	Wees realistisch in uw ambitie.	BH/02, JS/RC01
AR2	ICT is geen 'quick fix' voor een probleem. De beslissers in het politieke veld geloven in ICT als ware het een wondermiddel voor het oplossen van allerlei beleidsvraagstukken.	JS/RC01, JS/RC02
AR3	Politieke deadlines kunnen fataal zijn voor een project.	BH/02
AR4	Er is (ook) bij ICT-ambities een kloof tussen beleid en uitvoering.	JS/RC01, JS/RC02
AR5	Heroverwegingen onderweg zijn vaak onvermijdelijk.	JS/RC22, LM/10
AR6	Een exitstrategie voorkomt doormodderen.	JS/RC22
AR7	Zorg ervoor dat u grip heeft op uw ICT-projecten.	BH/01, JS/RC15
AR8	De minister moet een volwaardige gesprekspartner zijn voor zowel de Tweede Kamer als de ICT-leverancier.	JS/PUBRC01, JS/RC16
AR9	Het is nodig dat besluitvorming gefaseerd plaatsvindt.	BH/03, JS/RC20, JS/RC22
AR10	Het is nodig dat beslissingen worden genomen op basis van goed onderbouwde plannen en projecten worden beoordeeld binnen het gehele projectportfolio (business case).	JS/RC01, JS/RC02
AR11	ICT-projecten van de overheid zijn vaak te complex.	JS/PUBRC03
AR12	Organisatorische complexiteit ontstaat omdat er verschillende autonome organisaties bij betrokken zijn.	JS/PUBRC01, JS/PUBRC06, JS/RC16
AR13	Afzonderlijke organisaties streven vaak eigen doelen na in plaats van gedeelde ketendoelen.	JS/PUBRC01, JS/PUBRC04, JS/RC01, JS/RC02
AR14	Bij het realiseren van politieke doelstellingen wordt vaak de relatie tussen ICT en organisatie over het hoofd gezien.	JS/PUBRC01, JS/RC01
AR15	Wat ICT-projecten vaak op een andere manier complex maakt dan andere typen projecten, is dat ICT- systemen relatief rigide zijn. Veel politieke en organisatorische processen zijn juist dynamisch en flexibel.	BH/04, JS/RC01, JS/RC02
AR16	Wijzigingen 'onderweg' zijn vaak technisch lastig te realiseren.	BH/04, JS/RC18, JS/RC19
AR17	ICT-projecten worden te complex door de combinatie van politieke, organisatorische en technische factoren.	BH/04, TAH/09
AR18	Bij complexe projecten is er geen balans tussen ambitie, beschikbare mensen, middelen en tijd.	BH/01, BH/04, JS/RC15
AR19	Goede informatie is noodzakelijk voor grip op (de uitvoering) van ICT-projecten. Dat geldt voor zowel de verantwoordelijke minister als voor de Tweede Kamer.	BH/03, NB/06
AR20	Ministers tonen graag daadkracht richting Tweede Kamer: daadkracht is zichtbaar door een ambitieus project te presenteren en daar een concrete, nobije deadline aan te verbinden.	BH/02
AR21	De ICT-leveranciers hebben voor hun voortbestaan opdrachten nodig, liefst grote.	BH/01, JS/RC01, JS/RC02, JS/RC20
AR22	De actoren hebben allemaal de 'natuurlijke' neiging te denken aan grote oplossingen voor grote problemen.	JS/RC01, JS/RC02, JS/RC09, JS/RC20
AR23	De actoren vormen geen tegenwicht voor elkaar.	JS/PUBRC01, JS/RC09
AR24	Een minister heeft beleidsinformatie nodig om goed te kunnen sturen.	BH/03, NB/06
AR25	Goede informatie is noodzakelijk om grip op (de uitvoering van) ICT-projecten te hebben.	JS/RC15, NB/06
AR26	De positie en verantwoordelijkheden van een minister verschillen voor projecten.	BH/01, JS/PUBRC01, JS/RC16
AR27	Het rapport over een ICT-investering moet een aantal voorgeschreven onderwerpen behandelen:	JS/RC01, JS/RC02
AR28	de relevantie van het nieuwe systeem voor de kernprocessen;	JS/RC01, JS/RC02
AR29	de samenhang met andere systemen;	JS/RC01, JS/RC02
AR30	de vraag of er geen beter - eventueel reeds bestaand – alternatief is;	JS/RC01, JS/RC02
AR31	bezinning op de bedrijfsprocessen om daar zo mogelijk verbeteringen in aan te brengen;	JS/PUBRC02, JS/RC02
AR32	een kosten-batenanalyse;	JS/RC01, JS/RC07, TAH/14, TAH/43
AR33	een beoordeling van het project ten opzichte van de totale IT-portfolio van de instantie.	JS/RC01, JS/RC02, TAH/77
AR34	Er is geen centrale administratie van ICT-projecten of van het mislukken daarvan.	JS/PUBRC01, NB/06
AR35	De ICT bij de politie behoeft bij verschillende onderdelen een verbetering van de programma-beheersing.	JS/RC01, TAH/77
AR36	Bij P-Direkt is de aanbesteding van de ICT-component mede door onvoldoende sturing mislukt.	JS/PUBRC01, JS/PUBRC04
AR37	ICT-systemen moeten vaak worden aangesloten op andere, vaak bestaande, systemen.	JS/RC01, JS/RC02, TAH/77
AR38	Een project kan hanteerbaar worden gemaakt door het op te knippen in kleinere, beter beheersbare deelprojecten.	JS/PUBRC03, JS/RC06
AR39	Elk van de actoren, ministers, Tweede Kamer en ICT-leveranciers, heeft, op zich om legitieme redenen, belang bij grote en ambitieuze projecten.	JS/RC01, JS/RC02, TAH/77

Figuur 1. Belangrijkste succes- en faalfactoren bij ICT-projecten volgens het rapport van de Algemene Rekenkamer

over de publicatie van Jan Oonincx: 'Het hoekje is, ongetwijfeld, gebaseerd op vele – voor 1982 – jaren ergernis, maar kan twintig jaar later bijna ongewijzigd in herdruk. Er zijn nauwelijks oorzaken weggenomen.'
Ook de Amerikaanse Standish Group (2003) doet al meer dan tien jaar onderzoek naar ICT. Zij richten zich met hun onderzoek nadrukkelijk op succes- en faalfactoren bij ICT-projecten. Hun onderzoek, dat heel toepasselijk 'Chaos' is gedoopt, verschijnt om de twee jaar. Hieruit blijkt dat in 2003 slechts 34 procent van de ICT-projecten is geslaagd, 51 procent niet volgens plan is verlopen maar uiteindelijk tot enig resultaat heeft geleid, en 15 procent volledig heeft gefaald. Recentere onderzoeken geven geen significant andere resultaten.
Uit de bovengenoemde onderzoeken en het rapport van de Algemene Rekenkamer blijkt dat anno 2008 het onderwerp nog steeds zeer relevant is. Het artikel 'Automatiseringsramp lijkt onvermijdelijk' (Dekker, 2007) kan zelfs alarmerend worden genoemd. Ook blijkt dat al in 1982 aandacht is gevraagd voor succes- en faalfactoren bij ICT-projecten.

Toetsing van bevindingen aan SUFFI-Chart

Conclusies van Algemene Rekenkamer en adviezen van Experts
Het is interessant om de casus van de Algemene Rekenkamer te toetsen aan de SUFFI-Chart. De conclusies van de Algemene Rekenkamer resulteren in 39 succes- en faalfactoren in figuur 1. In figuur 2 zijn 58 opmerkingen en adviezen van experts vastgelegd (Zaal&Toet, 2007/2008; Verhoef & Maas, 2008; Verhoef, 2007). Voor alle 97 items is nagegaan welke SUFFI's uit de SUFFI-Chart kunnen worden gerelateerd aan het betreffende item. Voor ieder item zijn een of meer gerelateerde SUFFI's opgenomen. Daarbij zijn de belangrijkste van toepassing zijnde SUFFI's weergegeven; er is niet gestreefd naar volledigheid.

Op de pagina hiernaast:
Figuur 2. Opmerkingen/adviezen van experts naar aanleiding van het rapport van de Algemene Rekenkamer

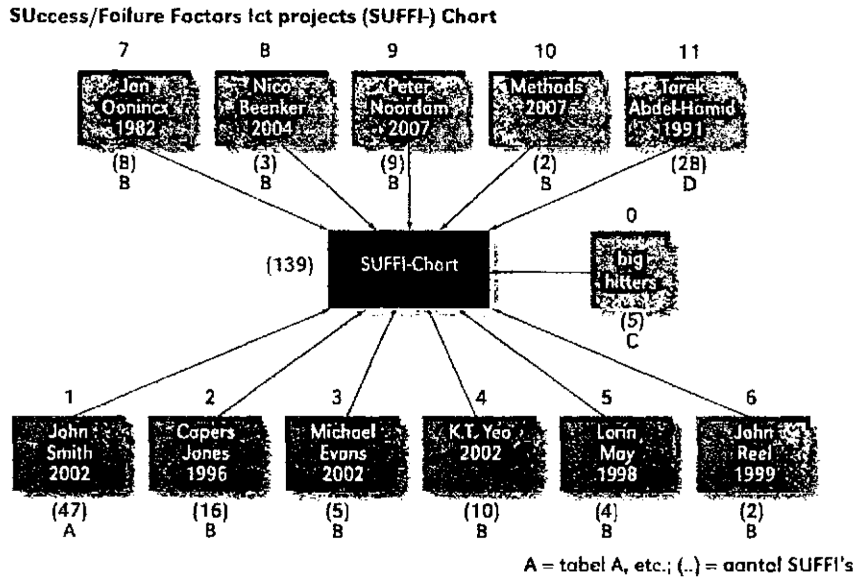
overzicht adviseurs/experts

afkorting	naam
CV	Chris Verhoef, hoogleraar informatica VU
DR	Daan Rijsenbrij, voorzitter Platform Outsourcing Nederland
GS	Gerard Sanderink, CEO Centric
HD	Hans Dieperink, hoofddirectie Inter Access
HN	Hans Nieuwenhuis, CIO Getronics PinkRoccade
JS	Jaap Schekkerman, oprichter online denktank IFEAD
JW	Johan van Wamelen, hoogleraar Centrum voor Publieke Innovatie, EUR
PG	Pieter Gremmen, senior projectmanager Twynstra Gudde
RHG	Rob van den Hoven van Genderen, docent ICT-recht VU

De SUFFI-Chart
De SUFFI-Chart (Van Dijk, 2008) is opgebouwd uit vier tabellen (zie figuur 3): tabel A bevat de SUFFI's die zijn aangedragen door publicaties van John Smith. Tabel B bevat de resultaten van een aantal internationale en Nederlandse onderzoeken. In tabel C zijn de succes- en faalfactoren opgenomen die door ten minste vier auteurs zijn genoemd als 'big hitter'. Tabel D bevat de resultaten van een onderzoek naar de op System Dynamics gebaseerde theorie van professor Tarek Abdel-Hamid van MIT (Abdel-Hamid & Madnick, 1991). De vier tabellen vormen tezamen het 'Reference model success and failure factors ICT projects', kortweg SUFFI-Chart. De SUFFI-Chart bestaat momenteel uit 139 SUFFI's. De SUFFI-Chart kan in de praktijk worden gebruikt en verbeterd door projectleiders en onderzoekers.

Conclusies
In de figuren 1 en 2 zijn de items (succes- en faalfactoren volgens de Algemene Rekenkamer en opmerkingen en adviezen van experts) gekoppeld aan SUFFI's van de SUFFI-Chart. Daarbij zijn de belangrijkste van toepassing zijnde SUFFI's weergegeven. In sommige gevallen is een een-op-een relatie van toepassing, bijvoorbeeld bij item AR3, 'Politieke deadlines kunnen fataal zijn voor een project', dat gerelateerd is aan SUFFI BH/02, 'Deadlines zijn niet realistisch'.
In andere gevallen is een item gerelateerd aan een combinatie van SUFFI's. Een voorbeeld is item AR7, 'Zorg ervoor dat u grip heeft op uw ICT-projecten', dat gerelateerd is aan de SUFFI's BH/01 ('Slecht projectmanagement') en JS/RC15 ('Slechte planning, management en uitvoering van project'). Hier wordt bedoeld dat als de SUFFI's BH/01 en JS/RC15 worden getransformeerd van faalfactoren naar succesfactoren, dat dan ook AR7 een succesfactor wordt.

nr.	opmerkingen/adviezen	auteurs	gerelateerde SUFFI's
	opmerkingen		
	Belangrijke punten die de Rekenkamer volgens de experts over het hoofd ziet:		
EX1	*Politiek onvermogen*: Kamer en regering hebben geen zicht op de impact en uitvoerbaarheid van de regels die ze verzinnen.	JS	JS/PUBRC01, JS/RC01, JS/RC05, JS/RC06
EX2	*Politiek onvermogen*: Onduidelijke processen, vage afspraken en compromissen kunnen niet worden geautomatiseerd.	JW	BH/04, JS/RC02, JS/RC04
EX3	*Gebrek aan transparantie*: Falende projecten worden met de mantel der politieke liefde bedekt.	JS	BH/03, JS/PUBRC04, NB/06
EX4	*Onvoldoende aandacht voor architectuur*: Architectuur is cruciaal om IT onder controle te krijgen en om zakelijk opdrachtgeverschap uit te voeren.	DR	JS/RC19
EX5	*Tekortschietend projectmanagement*: Uitloop is nooit de schuld van onervaren medewerkers, nieuwe technologieën en schuivende specificaties, maar altijd en uitsluitend van zwak management.	DR	BH/01, JS/PUBRC04, PN/10
EX6	*Tekortschietend projectmanagement*: Door tekort aan kennis en ervaring en onvoldoende probleemoplossend vermogen groeien kleine problemen uit tot grote problemen.	HD	BH/01, JS/RC23, PN/10
EX7	*Tekortschietende IT-deskundigheden*: Door een tekort aan technisch inhoudelijke kennis is er een te grote afhankelijkheid van externen, die veelal een eigen agenda hebben rondom de projecten.	HD, JS	PN/05, PN/10
EX8	*Opportunisme bij IT-bedrijven*: IT-leveranciers zijn te gretig om oplossingen te bouwen die eigenlijk zouden moeten worden afgeraden omdat de klant nog niet klaar is.	DR	JS/RC07, JS/RC12, PN/05
EX9	*Opportunisme bij IT-bedrijven*: Les één is om eerst te reorganiseren en dan pas te automatiseren. AvD: Al in het begin van de jaren zeventig werd dit door prof. Brussaard aan de TU Delft gedoceerd!	DR, HN	JS/PUBRC02, JS/RC02
EX10	*Onheldere allocatie van 'projecteigendom'*: Negatieve competentiestrijd, veelal van het soort 'daar ga ik niet over'.	HN	JS/PUBRC01, JS/RC04, JS/RC16
EX11	*Onheldere allocatie van 'projecteigendom'*: Te veel mensen bemoeien zich met de projecten, daardoor verzandt men in bijzaken.	GS	JS/RC02, JS/RC16
EX12	*Onheldere allocatie van 'projecteigendom'*: (Verborgen) politieke onwil tot samenwerking.	JS	JS/RC01, JS/RC02
EX13	*Toenemende juridisering van aanbesteding*: Opdrachtgever en opdrachtnemer hebben daardoor in de uitvoering onvoldoende speelruimte om af te wijken van de oorspronkelijke opdracht, ook al is dat voor de opdrachtgever objectief beschouwd beter.	HD	JS/RC08, JS/RC17
	adviezen		
EX21	Optimaliseer en vereenvoudig voordat je automatiseert eerst je werkprocessen.	JW	JS/PUBRC02, JS/RC02
EX22	Werk onder IT-architectuur: zorg voor samenhang, zowel op functioneel als technisch gebied.	JS	JS/RC19, TAH/77
EX23	Een goede IT-architect zet in op gevestigde 'best practices' en bewezen technologie.	DR	JS/RC19
EX24	De overheid moet wat IT betreft keuzen maken, kortom portfoliomanagement toepassen.	JS, CV	JS/PUBRC01, JS/RC01, JS/RC02, TAH/77
EX25	Portfoliomanagement kan het beste worden gebaseerd op de bedrijfseconomische maatlat (*net present value* en *return on investment*).	CV	JS/RC01, JS/RC02, JS/RC06, TAH/14
EX26	Informatietechnologie is duur, heel duur, omdat het allemaal handwerk is.	CV	JS/RC01, JS/RC02, JS/RC06, TAH/14
EX27	Een methodisch onderbouwde en economisch vertaalde schatting van het projectrisico is nodig.	CV	EV/07, JS/PUBRC05, JS/RC14
EX28	Durf mislukkingen tijdig te stoppen.	DR	BH/01, NB/05, NB/06, JS/RC22
EX29	Hoe groter het project, des te groter de kans op een mislukking. Alle risico's die normaliter min of meer te managen zijn, worden (bij het groter worden van de projecten) faalfactoren.	DR	JS/PUBRC03, JS/RC02, TAH/09
EX30	Projecten moeten minder omvangrijk worden.	Div	JS/PUBRC03, JS/RC02, TAH/09
EX31	Waar de ambities van de politiek niet zijn in te perken, zullen de bestelde projecten moeten worden opgedeeld in behapbare modules. De architectuurprincipes kunnen bij het modulariseren een leidraad bieden.	CV	JS/PUBRC03, JS/RC02, JS/RC19, TAH/09
EX32	De (rijks)overheid moet dubbel werk voorkomen.	DR	JS/RC19, TAH/77
EX33	Bewaak de scope van projecten, voorkom 'scope creep'.	Div	JS/RC18
EX34	Implementeer een overheidsbrede methode voor projectmanagement.	Div	BH/01, PN/08
EX35	Zorg voor een gecertificeerd systeem voor kwaliteitsmanagement.	Div	TAH/74, TAH/75, TAH/76
EX36	Entameer de mogelijkheid van overleg tussen leveranciers vooraf. Precompetitief overleg moet de norm zijn in plaats van uitzondering.	Div HD	JS/RC09, JS/RC31
EX37	Faalfactoren zitten meestal en zeker in de grootste mate bij de probleemstelling en de termijnen en budgetten.	HN	BH/01, JS/RC01
EX38	De overheid is verplicht contra-expertise te laten uitvoeren, maar dat gebeurt niet.	CV	JS/RC22, JS/RC30
EX39	Het is van groot belang regelmatig audits te laten afnemen door onafhankelijke deskundigen, ook tijdens de uitvoering. Dit is een taak voor de Rekenkamer.	Div JS	PN/14, JS/RC22
EX40	Leg projectrisico's bij de leveranciers. Van opdrachtnemers mag een ondernemende mentaliteit worden verwacht.	Div, HD	JS/RC09
EX41	Problemen bij grote automatiseringsprojecten komen altijd op hetzelfde neer: onvoldoende communicatie.	RHG	BH/03
EX42	De aanvrager weet vaak niet precies wat hij wil.	RHG	BH/04, JS/RC08
EX43	Om problemen te voorkomen is een langdurig voortraject nodig waaruit een gedetailleerd programma van eisen moet rollen.	RHG	JS/RC02, JS/RC08
EX44	Er moeten mijlpalen worden vastgesteld en er moet gekeken worden wanneer bepaalde deliverables geleverd moeten worden.	RHG	JS/RC15
EX45	De stuur- en overlegcommissies die de zaak in de gaten moeten houden, moeten deskundig zijn.	RHG	JS/PUBRC01, BH/03
EX46	Er moet een goede risicoanalyse worden gemaakt.	RHG	JS/PUBRC05, EV/07
EX47	Er was onvoldoende draagvlak in de organisatie.	RHG	JS/PUBRC01
EX48	Opdrachtgevers en projectleiders moeten niet alleen kijken naar het resultaat van het ICT-project, maar ook naar hoe de organisatie kan worden aangepast om met het project het hogere doel te bereiken.	PG	JS/RC01, JS/RC02
EX49	Er is een schromelijk gebrek aan professionaliteit in de ICT-wereld. Slechts een zeer klein gedeelte van de mensen die de ICT-projecten nu uitvoeren, is afgestudeerd in de informatica. Dat gaat toch nergens over!	CV	BH/01, PN/10
EX50	Eigenlijk zou de overheid alleen maar moeten werken met geaccrediteerde ICT'ers en niet met autodidacten.	CV	BH/01, PN/10
EX51	Er moet ook meer geld worden geïnvesteerd in softwareonderzoek.	CV	BH/01, PN/10
EX52	Universiteiten moeten mensen beter scholen in het managen en uitvoeren van grote ICT-projecten.	CV	BH/01, PN/10
EX53	Politieke deadlines zijn dodelijk voor IT.	CV	BH/02
EX54	Als je de doorlooptijd van een IT-project 'samenperst', zullen de kosten extreem veel hoger uitvallen.	CV	TAH/14
	Clinger Cohen Act (1996): acht minimale richtlijnen om kans te maken op funding!	CV	
EX55	Grote investeringen moeten de kerntaken van de organisatie ondersteunen.		JS/PUBRC02, JS/RC01, JS/RC02
EX56	Alleen investeren als er geen alternatief is, noch in de markt, noch bij een ander departement.		JS/RC01, PN/06
EX57	Eerst de werkprocessen optimaliseren, dan pas gaan automatiseren.		JS/PUBRC02, JS/RC02
EX58	Aantonen dat de verwachte return on investment voor wat je gaat doen beter is dan alternatieve inzet van publieke bronnen.		JS/RC01, PN/06
EX59	Het project moet consistent zijn met de federale en departementale informatiearchitectuur.		JS/RC02, JS/RC19
	risicobeperking		KY/03, EV/07
EX60	Geen geïsoleerde maatwerkoplossingen voor gedeelten van het project, omdat dat zeer slecht kan uitpakken voor het geheel.		JS/RC01, JS/RC02, JS/RC06
EX61	Gebruik volledig geteste pilots, simulaties of prototypes voordat je in productie gaat.		JS/RC32
EX62	Duidelijke targets en accountability over de voortgang.		JS/PUBRC04, JS/RC02
EX63	Verzekeren dat gebruikers meedoen, meebepalen en meedenken.		JS/PUBRC02
EX64	Investeringen moeten in kleine hapklare brokken gerealiseerd worden, in relatief korte tijd, en elk stukje op zich moet al bijdragen aan de kerntaken en een meetbaar positief nettoresultaat opleveren, dat onafhankelijk is van eventuele nog te bouwen stukken.		JS/PUBRC03, TAH/14
EX65	Gebruik een acquisitiestrategie die de risico's tussen overheid en aanbesteder redelijk en billijk verdeelt.		JS/PUBRC06, JS/RC09



succes- en faalfactoren die de 'big hitters' (BH) worden genoemd	door auteurs CJ	ME	KY	LM	JR	JO	NB	PN	totaal
Slecht projectmanagement (BH01)	+	+	+	+	+	+	+	+	8
Deadlines zijn niet realistisch (BH02)		+	+		+		+		4
Slechte communicatie (BH03)			+	+			+	+	4
Incomplete/zwakke definitie van requirements (BH04)			+	+	+			+	4
Onvoldoende betrokkenheid van toekomstige gebruikers (BH05)			+	+	+	+			4

Figuur 3. SUFFI-Chart: tabellen A, B, C en D geïntegreerd

In de figuren 1 en 2 zijn alle items gerelateerd aan één SUFFI uit de SUFFI-Chart of aan een combinatie van SUFFI's. Op basis daarvan hoeft de SUFFI-Chart niet te worden uitgebreid. Desondanks verdient het mijns inziens aanbeveling om nadere aandacht te vragen voor:

- AR33/EX24: De overheid moet wat ICT betreft keuzen maken, kortom portfoliomanagement toepassen;
- EX12: (Verborgen) onwil tot samenwerking;
- EX13: Toenemende juridisering van aanbesteding waardoor in de uitvoering onvoldoende speelruimte is om af te wijken van de oorspronkelijke opdracht, ook al is dat voor de opdrachtgever objectief beschouwd beter;
- EX49/EX52: Er is een schromelijk gebrek aan professionaliteit in de ICT-wereld. Slechts een zeer klein gedeelte van de mensen die de ICT-projecten nu uitvoeren, is afgestudeerd in de informatica. Universiteiten moeten mensen bovendien beter scholen in het managen en uitvoeren van grote ICT-projecten;
- EX59: Het project moet consistent zijn met de federale en departementale informatiearchitectuur.

Tot slot

Samenvattend kan worden gesteld dat anno 2008 het onderwerp SUFFI's nog steeds zeer relevant is. Uit de toetsing van de casus van de Algemene Rekenkamer blijkt dat de succes- en faalfactoren niet nieuw zijn maar voor een groot deel al jaren bekend. Toch besteden (project)managers bij ICT-projecten vooral doorgaans weinig aandacht aan SUFFI's. In het algemeen geldt dus: SUFFI's zijn bekend maar onbemind. (Project)managers die aan het begin van hun ICT-project één dag zouden besteden aan het serieus bestuderen van de SUFFI-Chart, zouden een aantal valkuilen kunnen voorkomen. Deze boodschap is overigens niet nieuw, ook de recensie uit 1982 eindigt met woorden van gelijke strekking.

Literatuur

Abdel-Hamid, T. & S.E. Madnick (1991). *Software Project Dynamics: An Integrated Approach*. Englewood Cliffs, NJ: Prentice Hall.
Algemene Rekenkamer (2007). *Lessen uit ICT-projecten bij de overheid – Deel A*. Den Haag, november 2007.
Berghout, E. (2002). Informatietechnologie, een bodemloze put? Intreerede Rijksuniversiteit Groningen, 22 oktober 2002.
Dekker, V. (2007). Automatiseringsramp lijkt onvermijdelijk. Trouw, 4 juni 2007.
Dijk, A.J. van Dijk (2008). *Some aspects of Project Management in ICT – Success and failure factors*. Proefschrift in voorbereiding.
Oonincx, J.A.M. (1982). *Waarom falen informatiesystemen nog steeds?* Alphen a/d Rijn: Samsom.
Standish Group International Inc. (2003). *Standard Group CHAOS Report*, 25 maart 2003.
Verhoef, C. (2007). Goed IT-bestuur begint bij minimale eisen. *Digitaal bestuur*, oktober 2007.
Verhoef, C. & J. Maas (2008). Overheid wil een iPhone voor prijs tweedehands Nokia. *Algemeen Dagblad*, 19 januari 2008.
Zaal, R. & D. Toet (2007/2008). Diverse artikelen in *Automatisering Gids* (Rolf Zaal) en *Computable* (Diederik Toet e.a.).

Ir. Aart J. van Dijk EMITA RE is sinds 1965 werkzaam binnen het vakgebied ICT. Hij is onafhankelijk IT-consultant/IT-auditor en werkt aan een proefschrift over ICT-succes- en faalfactoren in relatie tot zijn portfolio van ICT-projecten. E-mail: avede-info@planet.nl.

Publication 3

Role-based Access Control (RBAC)

Policy and implementation:

KLPD employee authorisation for using SAP R/3 applications

2004

Nothing is more difficult to prepare,
has such an uncertain chance of
success and is that risky to manage,
than the introduction of a new system.
Because the initiator experiences
animosity from all those that have
an interest in maintaining the
old system and only gains some half-hearted
approval from those that will benefit
from the new system.

Machiavelli [early in 1500]

This article previously appeared in:

IT Management [Select], September 2004, pages 31-47

Abstract
Our society is changing, and the Dutch police are changing with it. This creates new challenges for supporting the police using modern ICT resources. This is why the 'Police ICT Plan' was generated. ICT support is also being further professionalised with respect to police enterprise activities. One of the packages that have been used within the Dutch National Police Agency (KLPD) for several years is SAP R/3. SAP R/3 *Enterprise* was deployed in July 2003. On this occasion, the KLPD fully revamped their authorisation concept. This represented a response to the needs of the KLPD and the control bodies, such as the audit service of the Ministry of Internal Affairs, to arrive at a modern, insightful and manageable authorisation concept. The new authorisation concept is based on 'role-based access control'. There is considerable (international) interest in this method.
This article presents a summary of the ICT authorisation policy of the Dutch police and provides insight into how the authorisation concept is implemented in the KLPD, and in particular in SAP R/3.
The article concludes with a summary of experience acquired to date.

Authors
Aart van Dijk
Ton Algra

	Business	Information systems	Technology
Strategy			
Organization			
Operation			

Role-Based Access Control (RBAC) Policy and implementation: KLPD employee authorisation for using SAP R/3 applications

ABSTRACT

Our society is changing, and the Dutch police are changing with it. This creates new challenges for supporting the police using modern ICT resources. This is why the 'Police ICT Plan' was generated. ICT support is also being further professionalized with respect to police enterprise activities. One of the packages that have been used within the Dutch National Police Agency (KLPD) for several years is SAP R/3. SAP R/3 *Enterprise* was deployed in July 2003. On this occasion, the KLPD fully revamped their authorisation concept. This represented a response to the needs of the KLPD and the control bodies, such as the audit service of the Ministry of Internal Affairs, to arrive at a modern, insightful and manageable authorisation concept. The new authorisation concept is based on 'role-based access control'. There is considerable (international) interest in this method.

This article presents a summary of the ICT authorisation policy of the Dutch police and provides insight into how the authorisation concept is implemented in the KLPD, and in particular in SAP R/3. The article concludes with a summary of experience acquired to date.

1 Introduction

1.1 Information provision in the police corps

Our society is changing, and the Dutch police are changing with it. One of the current challenges is working cooperatively on the security and quality of life of a specific neighbourhood, district or municipality. Computers are useful tools for supporting this cooperation. Thanks to computers, we can send e-mails and faxes, consult databases and search engines, and create electronic files. We can quickly and conveniently communicate and exchange information – within a police corps, among corps, and with chain partners.

In practice, a few things leave something to be desired. One reason is that the police corps use different programs, which complicates exchanging information and employees. As a result, it sometimes seems that ICT does not adequately support police work, or sometimes even hinders it. For this reason, the three police consulting groups launched the ICT Platform in 1998 and the ICT Steering Group in 1999. The ICT Steering Group generated a phased plan called the 'Police ICT Plan'. This plan is aimed at revamping the ICT organisation without hampering routine police activities. In early 2001, the steering group and the three police consulting bodies approved the Police ICT Plan. The ICT project has since been initiated and will continue until sometime in 2005.

Police ICT Plan

The Police ICT Plan describes the organisation of the revamped ICT in broad terms. The primary principles of this plan are:

1. ICT is henceforth a common issue. The individual corps now often act as 'ICT islands' that set their own courses, but ICT is something that belongs to and serves the entire police organisation.
2. In the revamped ICT structure, demand and supply will be clearly separated. The two sides will be represented by a 'customer organisation' and a 'supplier organisation'. Both of these are group-level organisations, and thus represent *all* of the police corps.
3. ICT is steadily becoming more complex. It is thus highly important to raise the knowledge level within the organisations. All users must in fact be able to use computers (even) better.

Customer organisation

The Police ICT Plan provides for a 'customer organisation'. It consists of the sections in the various corps that are responsible for information management, along with the group-level Police Information Management Collaboration (*Coöperatie Informatiemanagement Politie*, or CIP for short) organisation. A collaborative organisation was chosen because the police corps explicitly cooperate in the CIP and jointly guide this cooperation.

Supplier organisation

The Police, Judiciary and Security ICT Services Collaboration (ISC) acts as the ICT service provider. It satisfies the wishes of the 'customer' by providing suitable ICT solutions or supplying the proper products. ISC will act as an intermediary, with products being purchased in the retail market. The ISC can also offer its own products and services. The ISC cooperates closely with the Information and Technology Organisation (ITO), an agency of the Ministry of the Interior and Kingdom Relations (MIKR) that provides ICT services to the police, judiciary and other chain partners. This close cooperation between the ISC and the ITO leads to a fusion of the ISC and ITO. The ISC is (indirectly) controlled by the corps and the two police ministers.

SAP R/3

The SAP (Systems, Applications and Products in Data Processing) R/3 package is an important ICT application used in the enterprise activities of the Dutch police. This package is used by the National Police Agency, among others. Access security for SAP R/3 (or SAP for short) is a significant concern of the KLPD.

1.2 Organisation of the police and the KLPD

The Dutch police are organised as 25 regional police corps and the National Police Agency (*Korps Landelijke Politiediensten*, or KLPD for short). The KLPD is a special corps with central staff offices and group departments, including the Group Department for Information Provision (CDI), as well as a fascinating collection of enterprise service departments, including the National Investigation Department, the Aviation Police Department and the Traffic Police Department.

2 SAP R/3 in the KLPD

SAP R/3 is used as a 'best of breed' package by the KLPD in the financial, logistics, personnel, and general and technical services areas. In addition, several new applications have been generated, such as 'aircraft fleet maintenance', 'forensic weapons', 'salary processing', and 'electronic shopping for the Logistics Department'.

Other applications, such as Employee Self Service (ESS), Customer Relation Management (CRM), Document/Record Management Systems (DMS), and Web Application Server (WAS), also form part of the picture. In summary, it can be said that SAP R/3 is a very important package for the enterprise activities of the KLPD.

3 Authorisation within the Dutch Police

3.1 Dutch Police Basic Security Level (BBNP)

The BBNP (*Basisbeveiligingsniveau Nederlandse Politie*) addresses basic security requirements and measures. The basic security measures are the measures employed for information security in each of the corps and the supraregional information systems.

The Police Information Security Regulations

(*Regeling Informatiebeveiliging Politie*, or RIP for short) define information security as the *reliability of information provision*, which is regarded as a *component of quality assurance* for the enterprise processes and underlying information systems. The BBNP guideline is thus *aimed at the quality of the information provision process* and defines a minimum set of measures for ensuring the reliability (i.e., the availability, integrity and exclusivity) of the information systems.

3.2 Authorisation guide

Corps employees are granted specific authorisations to allow them to use information systems and the information in them. These authorisations are dependent on the *function* or *functional role* performed by the employee. An employee only receives access to information systems, and may only exercise authorisations with respect to the data stored in these systems, to the extent that this is necessary for his work (*'need to use'* and *'need to know'*). Before an employee can process data, certain things must be *arranged* on the user organisation side and certain things must be *configured* on the automation organisation side.

Separation of duties

There must be a *separation of duties* in granting, modifying and revoking authorisations. It is undesirable for a person to be able to grant himself or someone else authorisations without external control. The various steps in the authorisation process must therefore be assigned to persons holding different positions. The following steps are distinguished in the guide:

- *Disposition task*
  This task lies with the person or persons who have been empowered to grant authorisations for using an information system and the information in it.
- *Registration task*
  This task lies with the person responsible for assessing the granted authorisations based on the defined conditions and handling the granted authorisations. He maintains a central registry for this purpose.
- *Execution task*
  This task lies with the person or persons who actually use the granted authorisations in the information system.
- *Control task*
  Routine supervision of the use of granted authorisations is the responsibility of the manager of the organisational unit where the employee is located. In addition, the corps manager can periodically have audits performed by an independent (external) EDP auditor.

Authorisation procedure

It is essential to have an agreed *authorisation procedure* for granting, modifying and revoking authorisations in writing, and an *authorisation authority* that supervises proper operation of the procedure. This authority performs the following activities at minimum:

- Ensuring that authorisations are granted in accordance with the legal framework and the conditions specified by the system owner.
- Periodically investigating whether the list of persons allowed to grant authorisations is still correct.
- Periodically checking whether the granted authorisations still correspond to the actual situation.
- Recording the granted authorisations and maintaining these records (registration task).
- Reporting regarding this set of activities to the corps chief, if he is tasked with executing the authorisation procedure, and otherwise to the corps manager.

Conditions

Authorisations are granted under the following conditions:

- Authorisations are requested and granted *in writing*.
- The function and/or functional role of each employee has been defined and is in effect.
- There is a current job description for each employee.
- The granted authorisations correspond to the tasks stated in the job description.
- The person granting the authorisation is authorised to do so.
- The duration of the authorisation is individually

specified for each authorisation.
- The authorised person has completed the necessary training for using the information system.
- The employee is actually an employee of the corps.
- The necessary security investigation of the authorised person has taken place.
- The user ID of the authorised person is correct.
- Authorisations are granted in accordance with the measures in the BBNP guideline.

4 Role-Based Access Control (RBAC)

The concept of role-based access control (RBAC) plays an important role in access security. This is a methodical approach originating from the US National Institute of Standards and Technology (NIST). The motive for developing RBAC was the desire to have an access security method that is suitable for enterprise applications and that simplifies the administration and maintenance of access security. After international agreement on the RBAC standard was achieved in 2002, many producers have based their access security software on it.
RBAC is based on *roles*, which are standardised sets of functions that are suitable for multiple users. Using functions and standardised sets of function linked to them is not new (see van Dijk, 1994), but a large amount of literature on the subject of RBAC has appeared in recent years, such as the book *Role-Based Access Control* by several employees of the NIST. RBAC is briefly described in this section.

In RBAC, the entities *users*, *subjects*, *objects*, *operations* and *permissions* play a prominent role, as do the relationships between these entities. *Users* are the entities that use an information system. A *subject* is a computer process or program that performs actions on behalf of a user. These actions are performed on an *object*, which is a resource that is accessible to the computer system. An *operation* is an action performed by a subject. *Permissions* (which are also called *privileges*) are authorisations to perform actions. Such an action involves a combination of an object and an operation.
Figure 1 shows the relationship between users, roles, permissions, operations and objects.
The 'subject' entity is not discussed in this article.

In RBAC, it is thus not allowed to link permissions directly to users. Permissions are granted by means of two links: *a link between users and roles* and *a link between roles and permissions*. A significant advantage of this arrangement is that if a user is assigned a different position in the enterprise, only the user–role link has to be changed. This does not require a large amount of specialised knowledge.
There are also major advantages with regard to disposition and control activities. As the number of roles is limited and mirrors the organisational structure, the number of changes will be small after an initial period. In addition to yielding considerably lower costs, using approved roles provides continual overview and insight into authorisation activities.
RBAC applications can be found at many different levels in ICT, such as in operating systems, database management systems, networks, workflow systems and Web services.
RBAC can also be used in Single Sign-On situations.

5 Authorisation in the KLPD

Conclusion

Based on the above (see also Figure 2), it can be concluded that:
1. *The 'need to know' and 'need to use' principles must form the basis for granting permissions or authorisations with respect to data, information and information systems.*
2. *Permissions or authorisations with respect to*

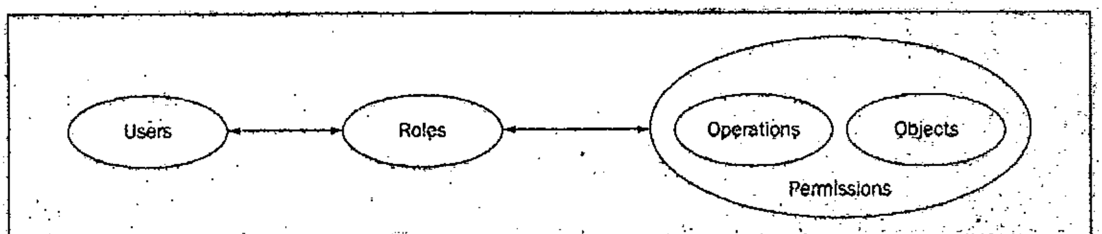


Figure 1. Relationship of RBAC components

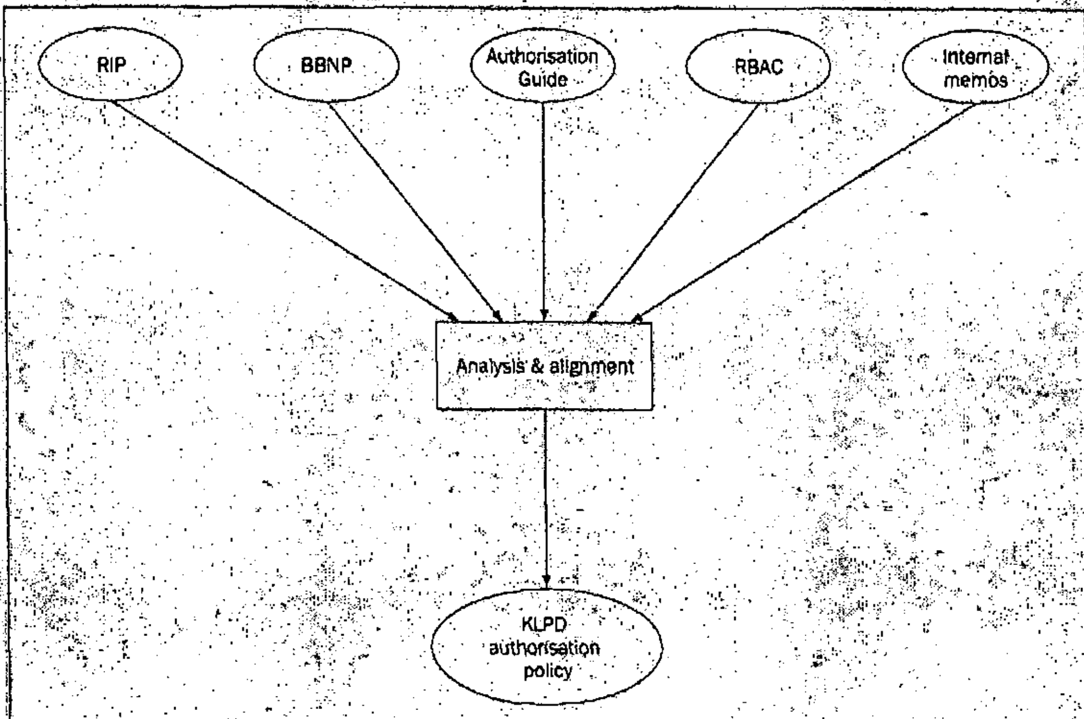


Figure 2 Generating a KLPD authorisation policy

data, information and information systems must be granted using two links:
- *a link between users and functions (or functional roles);*
- *a link between functions (or functional roles) and permissions (or authorisations).*

3. *Directly linking users to permissions or authorisations is not allowed.*

6 Authorisation in SAP R/3

Many software suppliers have responded to the international interest in role-based access control. RBAC can also be used in SAP R/3 Enterprise. SAP R/3 uses roles and can combine roles to form composite roles. In RBAC, the term 'role' is primarily used in the functional sense. In SAP R/3, the term 'role' refers to a technical system role. In the KLPD SAP R/3 authorisation concept, the terms function and functional role (which have a process-related nature) were chosen for the KLPD business view, while the terms SAP R/3 role and SAP R/3 composite role (which relate to technical aspects of the system) were chosen for the SAP R/3 system view.

The relationship between the business view and the system view is defined by a 1-to-1 link between a function or functional role and a SAP R/3 composite role (see Figure 3).

Tasks are distinguished within a function or a functional role. SAP R/3 roles are clustered in a SAPR/3 composite role (see Figures 3b and 4). The actual link is made inside SAP R/3 between the user ID and the SAP R/3 composite role. In this way, a user is linked based on his function or functional role (role-based).

The authorisation structure within RBAC and SAP R/3 is designated as 'object oriented'. An authorisation object forms the basis for determining the access privileges. It has at most ten fields for this purpose, which are defined in the data dictionary. The authorisation objects and associated authorisation fields and activity codes (values) are linked to a role via a profile (a set of authorisation objects). Besides authorisation objects, transactions can also be coupled to a SAP R/3 role via a menu (see Figures 4 and 6).

Generic and derived roles

Certain positions occur relatively frequently in a company or organisation. An example is the position of sales manager. Each sales manager then has a specific area of responsibility. Within KLPD, there is a similar situation for the position of personnel officer. Each personnel officer is responsible for one or more departments.
In SAP R/3, it is possible to use 'generic roles' and 'derived roles'. A role can be described using a generic role. The *derived* roles are derived from the generic role. They differ from the generic role by having different entries in the 'organisational unit' field.
Derived roles can easily be derived from a generic role. If the generic role is modified, the derived roles automatically receive the same modifications. This simplifies role maintenance and ensures consistency.

7 SAP R/3 Authorisation in the KLPD

7.1 Introduction

In Section 5, it was concluded that permissions or authorisations with respect to data, information an information systems must be granted using two links:

- a link between users and functions or functional roles;
- a link between functions or functional roles and permissions or authorisations.

The implementation of RBAC in SAP R/3 Enterprise is briefly described in Section 6. In this section, we describe how all of this is used in the KLPD (see Figure 5).

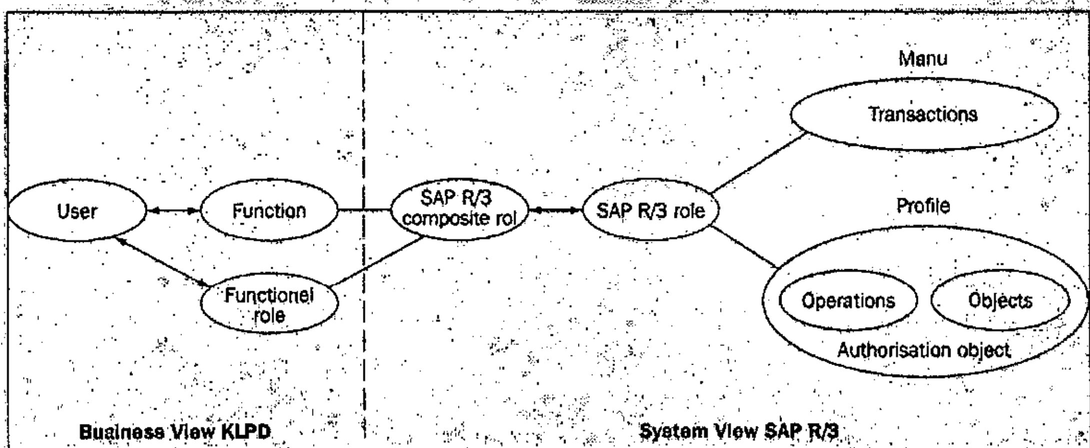


Figure 3a. Relationship between the KLPD and SAP R/3

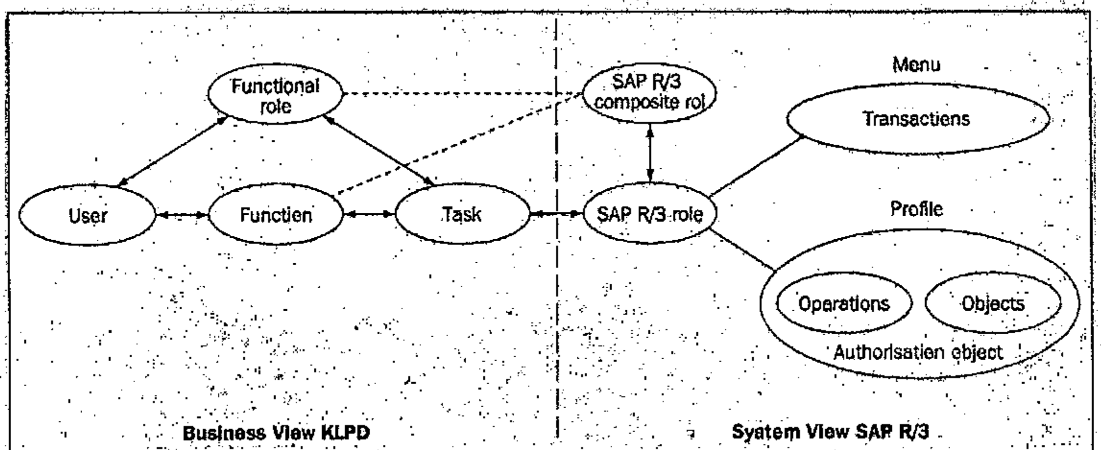


Figure 3b. Relationship between the KLPD and SAP R/3

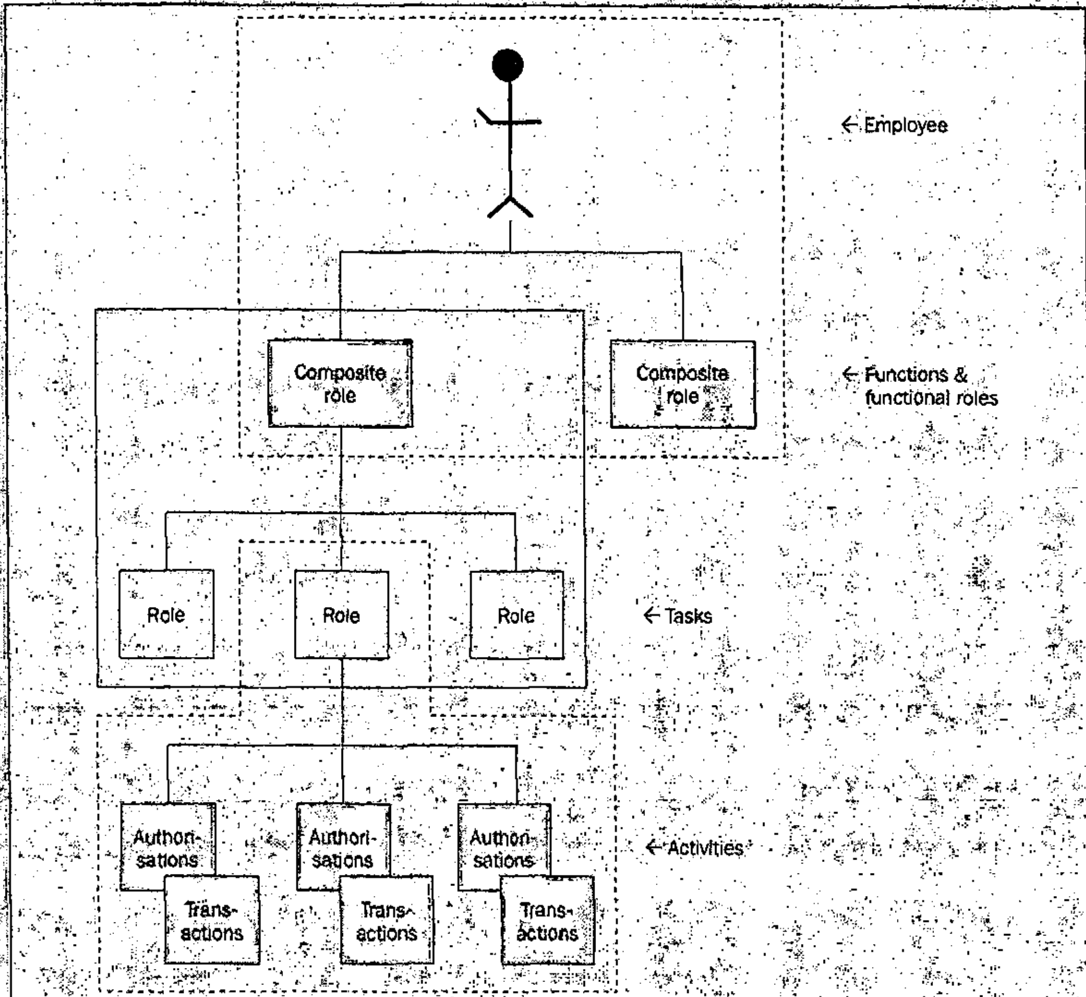


Figure 4. Relationship between the KLPD and SAP R/3

7.2 Roles and composite roles

Figures 3a, 3b and 4 show the relationship between functions and tasks in the organisation described in the Administrative Organisation (AO) and SAP R/3. Functions have a permanent nature. Functional roles have a process-related nature. SAP R/3 works with composite roles and roles. In practice, tables or matrices of composite roles versus roles are prepared. Based on the description of the function and associated tasks, the composite role and roles are generated by Application Management via Functional Management. Figure 6 shows an example of a function and its ultimately associated authorisations for a Personnel Management Assistant in the Water Police Department.

7.3 Implementation approach

When starting to implement RBAC, it is very important to define the functions and functional roles and the authorisations necessary for each function or functional role. Here a distinction can be made between different methods for developing functions:

- The *top-down* or *greenfield* method. This consists of using the organisational structure, security policy, job descriptions and process descriptions to arrive at an acceptable level of detail in defining the functions for using SAPR/3, and thereby the necessary authorisations.
- The *bottom-up* or *role-mining* method. This starts with existing authorisations and uses interviews with managers to determine which functions exist within a department or team.

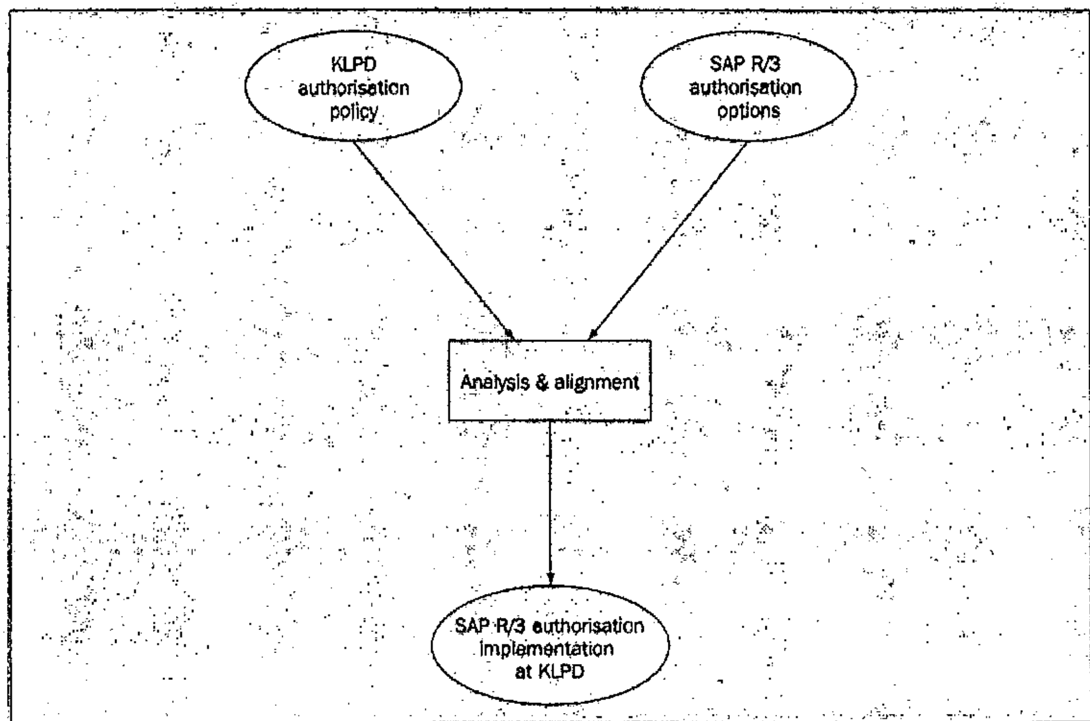


Figure 5. Alignment of KLPD policy with SAP R/3 authorisation options

A hybrid approach is also possible, consisting of a combination of the top-down and bottom-up methods. Two types of hybrid approach can be distinguished:

- *Type 1:* some functions or functional roles are developed using the one method, and others are developed using the other method.
- *Type 2:* functions and functional roles are initially determined using the bottom-up method, and then modified using insights obtained from the top-down method.

A hybrid approach employing both Type 1 and Type 2 was used by the KLPD to introduce the authorisation concept into SAP R/3. This was necessary due to the different initial positions of the various SAP applications. Type 1 was used for the personnel and logistics modules. Type 2 was used for the financial modules.

Personnel
The personnel modules of SAP 2003 were deployed in the second quarter of 2003. This was a greenfield situation, which formed an excellent starting point for the top-down method (much preparatory work and little tidying up).

Logistics
The functions were charted by starting with the actual situation and combining it with the common-sense method. After that the peep effect was utilised (little preparatory work and much tidying up).

Finance
Excellent results were achieved thanks to good interactive cooperation between Application Management and Functional Management.

Once the authorisation concept had been introduced, the authorisations in the various modules were reasonably well standardised. After this, a phase was started in which the structures of the roles were fine-tuned to yield a truly unified authorisation concept.

7.4 Separation of duties

The various steps in the authorisation procedure are assigned to persons holding different positions:

- *Disposition task*
  This task lies with the persons who have been empowered to grant authorisations for using an information system and the information in it. This involves the heads of the FBA, P&O and Logistics departments.
- *Registration task*
  This task lies with SAP Functional Management, which is tasked with assessing the granted authorisations based on the defined conditions (signalling task) and handling the granted permissions (this does not apply to Functional Management's own authorisations). It maintains a central registry of this. FM also has an advisory function.
- *Execution task*
  This task lies with Application Management, which actually implements the authorisations in SAP R/3.
- *Control task*
  Day-to-day supervision of the use of granted permissions is the responsibility of the manager of the organisational unit where the employee is located. Some of the principles here are:
  - If more than one function or functional role is assigned to an employee, a check must be made to see whether this results in an undesired combination of functions.
  - For new or modified functions, a check must be made to see whether the function contains any undesired combination of tasks. A transaction matrix of undesired combinations can be helpful for this.
  - For new or modified SAP R/3 roles and SAP R/3 composite roles, a check must be made to see whether this generates an undesired combination of authorisations.

Besides this, it is the task of the 'authorisation authority' to monitor proper operation of the authorisation process. The corps manager can periodically have audits performed by an independent EDP auditor. The Audit Department of MIKR performs audits at set times. Adequate reporting options are of great importance with regard to the control activities.

7.5 Quantitative data

Table 1 lists some quantitative data related to the authorisation concept as implemented in SAP R/3 at KLPD.

SAP entity	Qty
Composite roles	146
Generic roles	17
Derived roles	195
(Individual) roles	68
Authorisation objects	1305
Authorisation fields	80 671
Users	422

Table 1. Quantitative data, December 2003

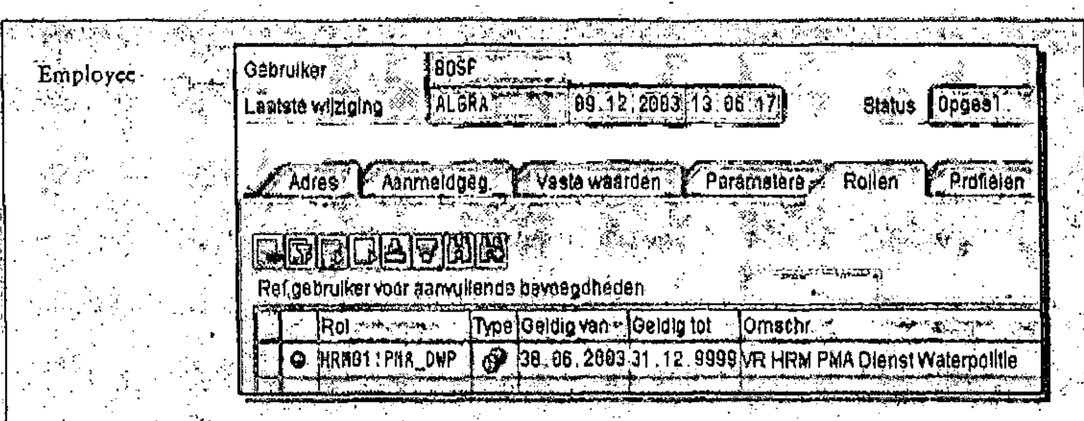


Figure 6. (zie vervolg)

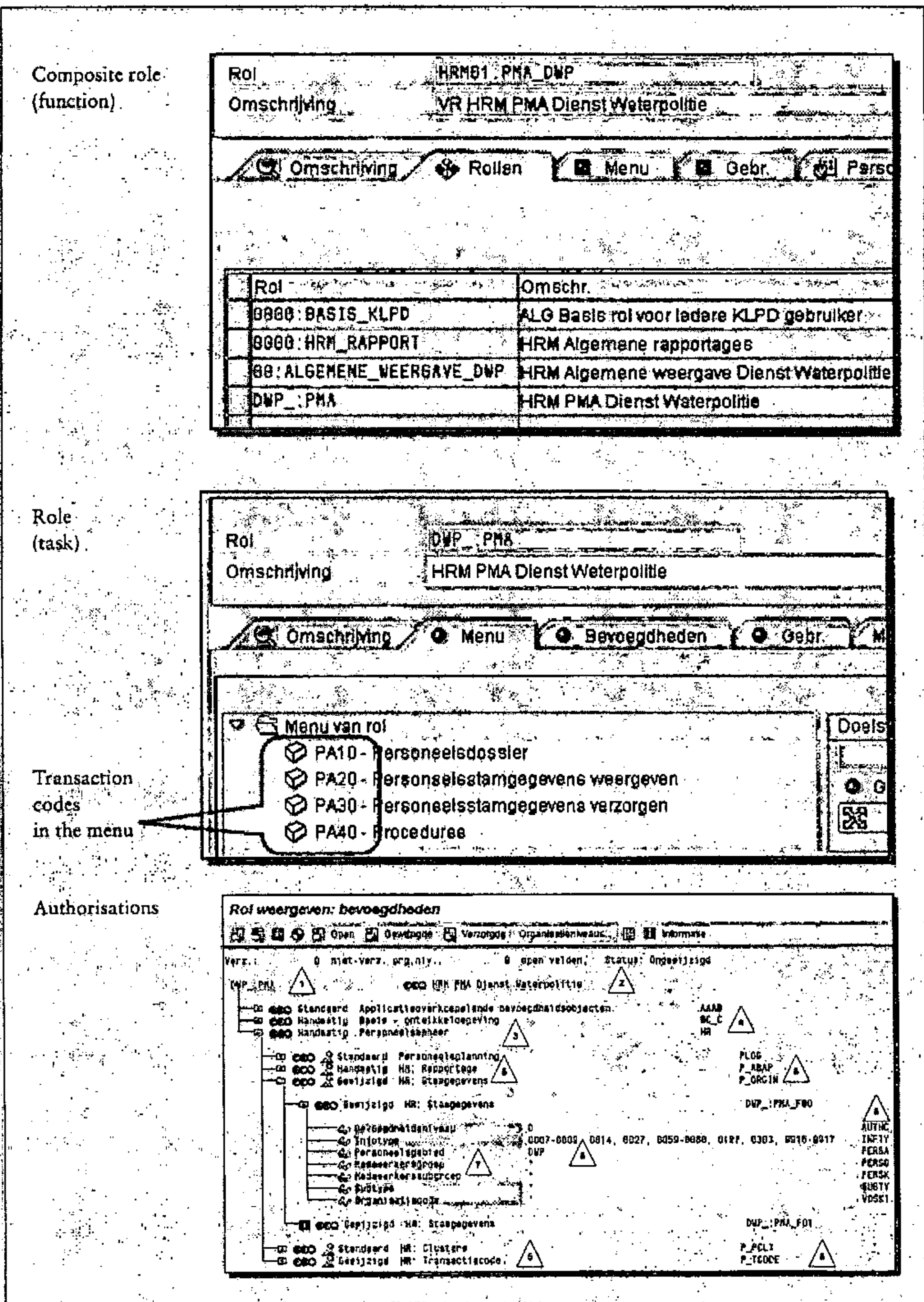


Figure 6 (zie Vervolg)

No.		Description
1	Role	Technical name
2	Role	Description
3	Authorisation objects	Names
4	Authorisation objects	Technical names
5	Objects	Descriptions
6	Objects	Technical names
7	Object fields	Descriptions
8	Object fields	Technical names
9	Object content fields	Authorisation values or operations

Figure 6. Example of a (single) function and its associated authorisations

8 Experience

8.1 Introduction

Configuring logical access security and authorisations in SAP R/3 is no easy task. From experience, it is evident that putting an authorisation concept into practice is often accompanied by complaints regarding authorisations. KLPD's experience in introducing the new authorisation concept based on functions and roles is described in Section 8.2. Experience from elsewhere is described in Section 8.3.

8.2 KLPD experience

General

- There is a (natural?) tendency to assign personal functions and roles. Although the literal sense of the model is followed, the underlying philosophy and principles are not followed.
- The chosen method provides a good opportunity for separation of the duties of FM and AM.
- No AM actions are necessary when people change positions, so the number of technical changes is much smaller.
- The authorisation concept gives auditors and administrators a considerably better view of granted authorisations than before.
- The implementation was performed using a combination of the actual situation and the common-sense method. This allowed a start to be made (see *Compact*, 2003/1). After that, a top-down stage was necessary for optimisation.

Logistics

- During the implementation of SAP R/3 Enterprise, the employees of the Logistics Department were confronted with many authorisation changes, which caused work processes to stagnate.
- As the Logistics Department did not provide sufficient input, the authorisation concept for the Logistics Department was implemented according to the Administrative Organisation.
- During the initial weeks after the implementation, (provisional) ad-hoc makeshifts were used to allow work to continue. The primary reason for this was that the authorisation concept was not adequately tested, due to employee absences in the Logistics Department.
- The Logistics Department desires to achieve a situation in which process responsibilities are assigned to a single position and the departments cooperate more on getting the entire job done.
- To prevent authorisation problems from occurring, end users must not be given (excessively) broad authorisations.
- SAP Application Management desires to transition to a (more) manageable authorisation concept and implementation that is easy to maintain and easy to understand for the KLPD as well as the MIKR Audit Department.
- The end users have been informed that the Administrative Organisation and authorisations will be restructured. This may means that in the near future, certain tasks must be done by someone else in the organisation.
- It will be indicated which information must be

classified in connection with an extra security level for weapons and munitions.

Finance

- FEA experienced the combination of introducing a new release of SAP R/3 (Enterprise) and a new authorisation concept as a severe strain ('it was made rather difficult for both sides'). As a result, a number of authorisation requests were put aside at the beginning.
- The new authorisation concept has produced increased separation of duties.
- The working method of Functional Management has changed. The new authorisation concept demands more 'think before you act'.
- Functional Management is obtaining better insight into SAP R/3.

Personnel

- Employees of the Personnel Department are only authorised to access data for the employees in their department. In some cases, employees are authorised for several departments. This causes problems when an employee changes departments. One solution would be to use a dummy department. If all relevant personnel employees have access to this dummy department, a relocation can be made via the dummy department. For a relocation from department A to department B, the employee in department A can move the relevant data to the dummy department. The employee in department B can them move the data from the dummy department to department B.

  Presently, relocations are performed by a central personnel staff member. An argument for continuing to use this method is that using a dummy department is cumbersome and requires absolute precision. For a central staff member, relocations will be a routine task, but for local personnel employees they will be infrequent events.
- If an employee is moved from department A to department B, the historical data (the data prior to the time of relocation) are not accessible to the personnel employee in the new department. If necessary, the personnel employee in department B will request information from his colleague in department A.
- The policy states that all employees have only one function, but exceptions can be made for good reasons. This will certainly be the case for several employees in the Personnel Department.
- An authorisation policy is needed.
- There will be a new function for the Training and Event Management (TEM) module. The employees in question are not permitted to have access to weapons data and the Special Investigation Applications Department.
- Employees 'can do what they have to do'.
- It is not absolutely necessary to technically enforce all authorisations; procedural agreements can be adequate in some cases. The challenge is to find a good balance. This can be determined using a risk analysis. With regard to screening off the Special Investigation Applications Department (DSRT) and the National Investigation Department (DNR), technically enforced authorisations are necessary.
- *The implemented authorisation concept is a very good system. One of its positive effects is that the organisation increasingly thinks in terms of functions instead of persons.*

Functional Management

- *The authorisation concept is clear.*
- The decision to use a 'general display role' was a good choice.
- The chosen nomenclature is not always the best; a few adjustments would be desirable.
- One of the composite roles caused problems, but that has been solved.
- Where will the control body be positioned, and what will be its tasks?
- Will there be separate start menus for new projects, such as Contracts and FLIMS (Flight Maintenance of the Aviation Police Department)?
- Who maintains and manages the matrices of composite roles versus roles?
- The request procedure for roles must be refined.
- Replacements and holidays must be taken into account when granting administration authorisations.

Application Management

- Using authorisations that are valid for only one or a few departments ('area of responsibility') creates *much work* and *restricts the operational flexibility* of employees.
- If a replacement is necessary for an employee due to illness or holidays, the replacement must

quickly have the new authorisations. If this does not work well, there is risk that people will use each other's user IDs and passwords.
- When restricted areas of responsibility are used, a *generic role* and the associated *derived roles* are used in the authorisation concept. Although derived roles are not supposed to be modified, this has occasionally happened. When a new generation of the associated generic role was created, the changes to the derived roles disappeared.
- Some employees find that screening off at the departmental level has gone too far and ask themselves 'Aren't we supposed to be a single organisation?'
- Although there is understanding for screening off (restricting access by colleagues) for a few departments, such as DSRT and DNR, people still wonder, 'How confidential can the DSRT data in SAP R/3 actually be?'
- Testing authorisations (particularly the negative tests) after changes takes a lot of time.
- Application Management must try to develop a method for formulating roles that causes roles to have a limited scope, allows them to be reused when formulating new functions, and avoids having to test them again when another role in the same function is modified.
- New functions should have to be approved by a control body. This acts to raise the threshold and prevents excessive function diversity.
- The KLPD has experienced various release changes. The profile generator tool has only been available in recent versions. Consequently, the SAP R/3 system in the KLPD had manually constructed profiles and authorisations as well as profiles and authorisations built using the profile generator. In a manner of speaking, the KLPD continued to further embroider at the level of release 3.0. With the change to the SAP R/3 Enterprise release and the new authorisation concept, the KLPD has again reached a 'state of the art' level with respect to authorisations in SAP R/3.

8.3 Experience elsewhere

KPMG

KPMG has a conducted a large number of SAP R/3 security studies in the form of audits and 'quick scans' in the area of logical access security. The SAP R/3 Security Competence Center has analysed the key aspects of the results of thirty-five studies. From theses analyses, it is evident that shortcomings were found in a large percentage of the organisations, ranging from the basic measures to configuration and management. These are often due to insufficient knowledge or a lack of attention. Based on this study, KPMG has prepared several tips (see Table 2).

9 Conclusions and recommendations

9.1 Conclusions

- *The implemented authorisation concept is considered to be a very good system.* One of its positive side effects is that the organisation increasingly thinks in terms of functions instead of persons.
- It is important to maintain strict separation of duties between Functional Management (registration function) and Application Management (execution function) with regard to authorisations for SAP R/3. The chosen method makes this quite feasible.
- No action by Application Management is necessary for job changes, so there are many fewer changes.
- The authorisation concept gives auditors and managers a much better view of the current situation that before. In a manner of speaking, the authorisation concept has changed from a 'black box' to a 'white box'.
- The configuration of the Administrative Organisation needs improvement. Beside further improvement of the processes, it is desirable to have a matrix of undesired combinations of functions and tasks.
- New functions should have to be approved by a control body.
- Using authorisations that are valid for only one or a few departments ('area of responsibility') creates much work and restricts the operational flexibility of employees.
- The administration functions related to authorisation must be honed.
- With the change to the SAP R/3 Enterprise release and the new authorisation concept, the KLPD has again reached a 'state of the art' level with regard to authorisations in SAP R/3. The experience gained with the new authorisation concept can also be used in other KLPD informa-

Tip	Description
1	Create authorisations for administrators as 'quick wins' using the principle 'all activities except...'.
2	Replace broad authorisations for developers and consultants with display authorisations.
3	Utilise the knowledge of your external advisor for configuring the relevant security parameters and filling in the tables.
4	Analyse the logging features in SAP so you can use them to best advantage.
5	Test the authorisations negatively as well as positively.
6	Select a structure for the authorisation concept that is sufficiently flexible, manageable, controllable and secure.
7	Use authorisation options as restrictively as possible.
8	Create a role-based configuration using flexible building blocks.
9	Isolate critical transactions functionally and technically into their own roles.
10	Give careful thought to clear nomenclature.
11	Also secure made-to-measure work.
12	Monitor the presence of inactive users and remove them in a timely manner.
13	Familiarise yourself with the Computer-Aided Test Tool (CATT) and use it for administration and configuration, for instance when uploading and downloading authorisations.
14	Establish clear agreements for making security impact analyses for change requests.
15	Analyse the utility and capabilities of the support tools.
16	Involve your own administrators in configuring the authorisation concept during the project.
17	Use Access or Excel with data from SAP tables to keep your documentation current.

Table 2. Tips from a KPMG study

tion systems and elsewhere within the police organisation.

9.2 Recommendations

- Provide good reporting capabilities with regard to granted authorisations.
- Further elaborate the AO procedures, including with regard to Functional Management and Application Management.
- Configure the control function, in addition to the disposition, registration and execution function.
- Investigate the need for restricting authorisations to the area of responsibility of personnel staff members.
- Try to achieve a good balance between technical and procedural measures, instead of attempting to enforce all authorisations by technical means.
- Give further attention to the administration functions.

10 Opinions on the New Authorisation Structure

The opinion of… Jacquelien Wijnhoud

Jacquelien Wijnhoud works in the Functional Management section of the Financial/Economic Affairs (FEA) group department. Her primary area of responsibility is Human Resources.
Jacquelien: 'Using roles and composite roles in SAP makes things clear, and so do the naming conventions that are used. With them, it's immediately clear which role you're looking at. The composite roles are grouped roles, so they give you a good overview of the functions available in a particular module.
I find it irritating that the user organisation regularly requests a new role or composite role without knowing what effect this will have in SAP R/3. People don't give sufficient consideration to other options, and once the role has been built, they realise it's not exactly what they wanted. The preliminary study is not adequate. This is especially true for projects that have a rather independent character, and where Functional Management is not sufficiently involved.'

The opinion of … Albert Aarts

Albert Aarts works in the Functional Management section of the Financial/Economic Affairs (FEA) group department. His primary area of responsibility is Finance.
Albert: 'It used to be simpler. We just passed the requests on to Application Management, who ensured that the new wishes were honoured. It worked, and no one looked to see whether users sometimes had authorisations that were too broad. In short, some users accumulated quite a few authorisations, and we lacked an overview. But the new authorisation concept based on 'need to know' and 'role-based' takes a bit of getting used to. There is more separation of duties, and that costs me quite a lot of time. I have to think more about something before doing it, and I sometimes run into problems when I'm surprised by an authorisation deficiency somewhere during the testing process. This is partly because some transactions pass data 'under the counter' to another transaction, which may be outside the granted authorisations. A big advantage is that I'm learning a lot more about SAP R/3, and that's really great!'

The opinion of ... Hans Smits

Hans Smits is the 'first among equals' in SAP Application Management.

'In contrast to the previous concept, the new authorisation concept is strongly based on functions within the organisation. That makes it possible to separate user administration from authorisation development. This produces a meaningful and logical separation of duties. The new authorisation concept was introduced at the same time as a new release of SAP R/3 (Enterprise). Although various things made strong demands on the (user) organisation, this did not cause any delays. By resolutely adhering to the new authorisation concept and using common sense in configuring the authorisations, we were able to exact the cooperation of the user organisation. Of course, that means that if the user organisation puts little effort into the new concept during the implementation phase, extra effort must be put into it during the maintenance phase. This could be clearly seen at KLPD. SAP HR was implemented in 2002, which already gave us a head start on the new authorisation concept. There were hardly any problems with this module with the release change. The authorisations related to the SD and MM modules had been regularly modified over the years and had become strongly person-specific. In that case, changing to function-related authorisation is no easy task. It demands a lot of understanding and patience from the implementer, as well as from the user organisation. As a result of adhering to the concept, the user organisation also started to think in a more function-oriented manner. Besides contributing to a better understanding of the authorisation concept, this also helps the KLPD in considering how to configure its processes and promotes 'chain awareness'.'

The opinion of ... Gé Kramer

Gé Kramer is the head of the Applications section, which is responsible for SAP R/3.

'Implementing an authorisation concept was a difficult issue from the very beginning of the SAP R/3 implementation. These capabilities were simply unknown to the organisation with the (outdated) applications it was using. Such advanced capabilities for separating duties were very limited at that time. The culture was also not always conducive to letting itself be forced into such a rigid scheme.
The KLPD was initially a merger of several highly different police activities that could not be placed in a regional organisation, and they were housed in several divisions.
A reorganisation was started in 2000, with twelve primary departments and four group departments being established in order to better fit with the organisation of the regional corps. This change had direct consequences for the match between the processes and the tasks for which people were made responsible.
During the most recent implementation of HRM, an activity was immediately identified for streamlining the processes and implementing the SAP configuration on that basis. The SAP authorisation concept, which is a science in itself with very many possibilities, was a direct aid in this process. Not everybody shared this opinion. Separating processes from person-specific tasks was a major change to the status quo. Because the implementation and actual maintenance fall within the ICT organisation, the attention to the authorisation concept quickly came to be seen as a new hobby of the ICT group. It is thus important for management to play an active role in implementing separation of duties based on the AO. The authorisation concept must therefore be a direct reflection of the AO agreed on by management.
Now that the authorisation concept is operational, practical experience shows that the theory is sometimes at odds with everyday reality. Fine-tuning after the implementation is thus also necessary, and here it is necessary to be alert to attempts to again implement person-specific authorisations.'

About the authors

Ir. A.J. van Dijk RE is an independent IT consultant and IT auditor employed by Avédé-info B.V., Zoetermeer.
He has conducted IT audits at the KLPD and is the project manager of the POTVIS project, which aims to improve the SAP R/3 infrastructure in the KLPD.

A.L.P. Algra is employed as a Senior SAP Consultant at Ordina Enterprise Applications. He has acted as a consultant for introducing and improving the authorisation structure in SAP R/3 for many companies, including multinationals. At the KLPD, he is involved in the POTVIS project, in particular the introduction of the new authorisation concept.

References

[Algr2003] Algra, Ton, SAP Autorisatie Concept, KLPD KLPD/Ordina, 2003 (internal)

[Baut2000] Bautz, J., e.a., Checklist informatiebeveiliging, Praktijkreeks informatiebeveiliging, Ten Hagen en Stam, 2000

[Dijk1994] Dijk, Aart J. van, Schermbeeldcommunicatie en autorisatie, Informatie, March 1994, vol. 36 no. 3, pp. 191–198

[ECIB2001-1] ECIB, Informatiebeveiliging Nederlandse Politie, Basisbeveiligingsniveau Nederlandse Politie (BBNP), ECIB, Den Haag, 13 December 2001

[ECIB2001-2] ECIB, Informatiebeveiliging Nederlandse Politie, Autoriseren, ECIB, Den Haag, 13 December 2001

[Ferr2003] Ferraiolo, David F, D.R. Kuhn and R. Chandramouli, Role-Based Access Control, Artech House, Boston/London, 2003

[Hael2002] Haelst, Werner Van and Jack van der Voort, Trends in het beoordelen van SAP R/3-toegangsbeveiliging, De EDP-Auditor, no. 1, 2002, pp. 11–18

[IBM2003] IBM Business Consulting Services, SAP Authorization System, Design and Implementation of Authorization Concepts for SAP R/3 and SAP Enterprise Portals, SAP PRESS, Waldorf, 2003

[Mien2003] Mienes, P. and B. Bokhorst, De (on)beheersbaarheid van toegangsbeveiliging, Compact, 2003/1, pp. 42–50

[Moss2002] Mossinkoff, O., AO/IC en autorisaties, Project Jurist 2002, Ministerie van Justitie, Den Haag, 2002

[Regi2001] Regieraad ICT, Bestek 2001-2005 Voor de vernieuwing van de informatiehuishouding van de Nederlandse politie, Driebergen, 16 February 2001

[Slui2000] Sluiter, J. and A.J. Vethman, Role based access control in e-business, Informatiebeveiliging, Jaarboek 2000/2001, pp. 44–49

[Review] Review
The concept of this article was reviewed by the following persons:
Willem van Amerongen (ICT Operations, Applications),
Gé Kramer (ICT Operations, Head of Applications),
Peter Regen (FEA, Head of Functional Management),
Hans Smits (ICT Operations, SAP Application Management),
Gerwin de With (FEA, AO/IC)

[Twee2003] Tweede Kamer der Staten-Generaal, Zicht op taakuitvoering politie, 28 791, Sdu Uitgevers, Den Haag, 2003

[Vree2001] Vreeke, A. and D.M. Hallemeesch, Richt jij de autorisaties even in?, De complexiteit van het SAP R/3-autorisatieconcept, Compact, 2001/6, pp. 27–33

[Wal2003] Wal, J.A. van der and N.L. Homma, Gegevensbeveiliging aan alle kanten lek, Automatisering Gids, 5 September 2003, p. 17

Publication 4

Gegevensuitwisseling (interfacing) van applicaties met behulp van een message broker

De applicaties van de Business Unit
BedrijfsCommunicatie van KPN Telecom

Interfacing applications by means of a message broker.
The applications of the Business Unit Business Communication of KPN Telecom

2001

Focus on fundamental solutions instead of focusing on symptoms and think long-term.

Bryan et al. [2007, Systems thinking]

This article previously appeared in:

IT Management [Select], April 2001, pages 37-55

Abstract
Business processes are increasingly supported by information systems that are part of the ICT. One will often ask oneself these questions: "How many and which information systems does the business / the organisation need?" and "Which data has to be exchanged between the information systems?". Data exchange often takes place using so-called 'one on one' links/interfaces. The data of one of the information systems is offered to the other information system and vice versa. Organisations with hundreds of information systems often have a large number of interfaces as well. That is also the case at the Business Unit BedrijfsCommunicatie (Business Communication) (BU BC) of KPN Telecom. The maintenance of all interfaces demands substantial personnel and financial effort. In order to make the management of the interfaces more controllable and in particular for reducing the costs, the BU BC decided to procure the Engin message broker. Using the message broker, a few things have been achieved. This article discusses the manner in which the message broker is applied and the experiences with it. Very recently, BaanERP was also connected to the message broker. This way, data exchange takes place between the business applications and BaanERP.

Aart J. van Dijk
Ruud E. Overeem
Johan J. Schijf

GEGEVENSUITWISSELING (INTERFACING) VAN APPLICATIES MET BEHULP VAN EEN MESSAGE BROKER

DE APPLICATIES VAN DE BUSINESS UNIT BEDRIJFSCOMMUNICATIE VAN KPN TELECOM

samenvatting

Bedrijfsprocessen worden meer en meer ondersteund door informatiesystemen, die onderdeel uitmaken van de ICT. Vragen die men zich vaak stelt zijn: "Hoeveel en welke informatiesystemen heeft het bedrijf / de organisatie nodig?" en "Welke gegevens moeten er tussen de informatiesystemen worden uitgewisseld?". Vaak vindt gegevensuitwisseling plaats met behulp van zogenoemde 1 op 1 koppelingen/interfaces. Gegevens van het ene informatiesysteem worden aangeboden aan het andere informatiesysteem en vice versa. Organisaties met honderden informatiesystemen hebben vaak ook een groot aantal interfaces. Ook bij de Business Unit Bedrijfs-Communicatie (BU BC) van KPN Telecom is dat het geval. Het onderhoud van alle interfaces vergt een omvangrijke personele en financiële inspanning. Om het beheer van de interfaces beter beheersbaar te maken en met name de kosten sterk omlaag te brengen heeft de BU BC twee jaar geleden besloten tot aanschaf van de message broker ENGIN. Met behulp van de message broker wordt een en ander bereikt. Dit artikel gaat over de wijze waarop de message broker wordt ingezet en de ervaringen daarmee. Zeer recent is ook Baan ERP gekoppeld aan de message broker. Op die manier vindt gegevensuitwisseling tussen de bedrijfsapplicaties en Baan ERP plaats.

1 Inleiding

1.1 Onderwerp van dit artikel

Bedrijfsprocessen worden meer en meer ondersteund met behulp van informatiesystemen, die onderdeel uitmaken van de ICT. Vragen die men zich vaak stelt zijn: "Hoeveel en welke informatiesystemen heeft het bedrijf / de organisatie nodig?" en "Welke relaties bestaan er tussen de informatiesystemen en welke gegevens moeten er tussen de informatiesystemen worden uitgewisseld?".

De afgelopen decennia was over deze vragen veel discussie. Tussen twee uitersten ontstonden veel tussenvormen. De uitersten waren enerzijds een groot aantal losse informatiesystemen die niet of nauwelijks met elkaar communiceerden en anderzijds het streven naar één allesomvattend informatiesysteem. De eerste vorm kan worden gekwalificeerd als "eilandautomatisering". Dit kwam in de zestiger en zeventiger jaren veel voor, maar ook heden ten dage is een dergelijke vorm van informatisering hier en daar nog waarneembaar. De tweede vorm was populair in de jaren zeventig. Veel informatici, met name zij die weinig daadwerkelijke praktijkervaring hadden, dachten met behulp van één geïntegreerd en allesomvattend, informatiesysteem alle informatieproblemen te kunnen oplossen. Dat ging echter volledig mis. De beoogde omvangrijke monolithische informatiesystemen kwamen niet van de grond.

De juiste oplossing is een beperkt aantal informatiesystemen. In hoofdstuk 2 wordt hierop nader ingegaan. Belangrijk daarbij is het uitwisselen van gegevens tussen de informatiesystemen. Vaak zijn dat zogenoemde 1 op 1 koppelingen/interfaces (zie figuur 3). Gegevens van het ene informatiesysteem worden aangeboden aan het andere informatiesysteem en vice versa. Organisaties met honderden informatiesystemen hebben vaak ook een groot aantal interfaces. Ook bij de Business Unit BedrijfsCommunicatie (BU BC) van KPN Telecom is dat het geval. Het onderhoud van alle interfaces vergt een omvangrijke personele en financiële inspanning. Om het beheer van de interfaces beter beheersbaar te maken en met name de kosten sterk omlaag te brengen heeft de BU BC twee jaar geleden de message broker ENGIN aangeschaft. Met behulp van de message broker wordt een en ander bereikt. Dit artikel gaat over de wijze waarop de message broker wordt ingezet en de ervaringen daarmee. Zeer recent is ook Baan ERP gekoppeld aan ENGIN en vindt gegevensuitwisseling tussen de bedrijfsapplicaties en Baan ERP plaats.

Dit hoofdstuk beschrijft de positie van de BU BC binnen KPN Telecom. In hoofdstuk 2 komt de probleemstelling aan de orde. Een korte schets van twee informatiesystemen wordt gegeven in hoofdstuk 3, terwijl het koppelen van applicaties uitgebreid aan de orde komt in hoofdstuk 4. De conclusie en toekomstige ontwikkelingen besluiten het artikel.

1.2 De positie van de BU BedrijfsCommunicatie binnen KPN Telecom

KPN Telecom is in Nederland een zeer belangrijke speler op de telecommunicatiemarkt en moet snel en flexibel inspelen op de dynamiek in de telecommunicatiemarkt en de steeds veranderende wensen van de klanten. KPN doet dat door een gedifferentieerd producten- en dienstenaanbod. Daartoe is de organisatie van KPN ingedeeld in Business Units, Verkoopeenheden en Productie Eenheden. Deze eenheden worden ondersteund door Staven, Professionele Service Eenheden en Gemeenschappelijke Service Eenheden. Met deze indeling wordt gestreefd naar een strakke sturing van businessdoelstellingen op verlies en winst en een heldere en eenduidige toekenning van verantwoordelijkheden en bevoegdheden.

Productie-eenheden zijn onder andere: het *Software Huis*, dat diensten levert aan de KPN eenheden op het gebied van applicatiesoftware (kennis, ontwikkelen, integreren, testen en beheer) en het *Datacenter*, dat diensten levert voor de implementatie, exploitatie en beheer van ICT-infrastructuur. Professionele Service Eenheden zijn onder andere: *Innovatie Unit ICT* (is gericht op innovatie van klantprocessen en leveringsprocessen), *Audit* (levert informatie en advies over de beheersing van de bedrijfsprocessen) en *KPN Research* (onderzoek en ontwikkeling op het gebied van technologie en sociale wetenschappen).

KPN Telecom omvat de volgende Business Units:
- Business Unit Carrier Services;
- Business Unit Vaste Telefonie;
- Business Unit Mobiele Telefonie;
- Business Unit Corporate Networks;
- Business Unit TeleCommerce;
- Business Unit BedrijfsCommunicatie.

1.3 De Business Unit BedrijfsCommunicatie

De BU BC houdt zich bezig met de inkoop, verkoop, installatie en service van alle apparatuur en systemen die bij klanten worden geplaatst. Daarnaast zorgt de BU BC voor het maken van aansluitingen in het aansluitnet. In het werkveld van de BU BC zijn zo'n 6000 mensen werkzaam: veel monteurs, planners en werkvoorbereiders maar ook marketeers, financiële medewerkers en personeelsfunctionarissen. Het doel van de BU BC is marktleider zijn en blijven en het realiseren van een goed rendement. Om dat doel te bereiken werkt de BU continu aan het verbeteren van de bedrijfsprocessen. Belangrijke zaken daarbij zijn het neerzetten en onderhouden van een heldere, effectieve en efficiënte organisatie en het bevorderen van het nemen van initiatief en verantwoordelijkheid van medewerkers naar klanten. De bedrijfsprocessen van de BU BC worden ondersteund door ICT. Dit is een taak van de Business Proces Group.

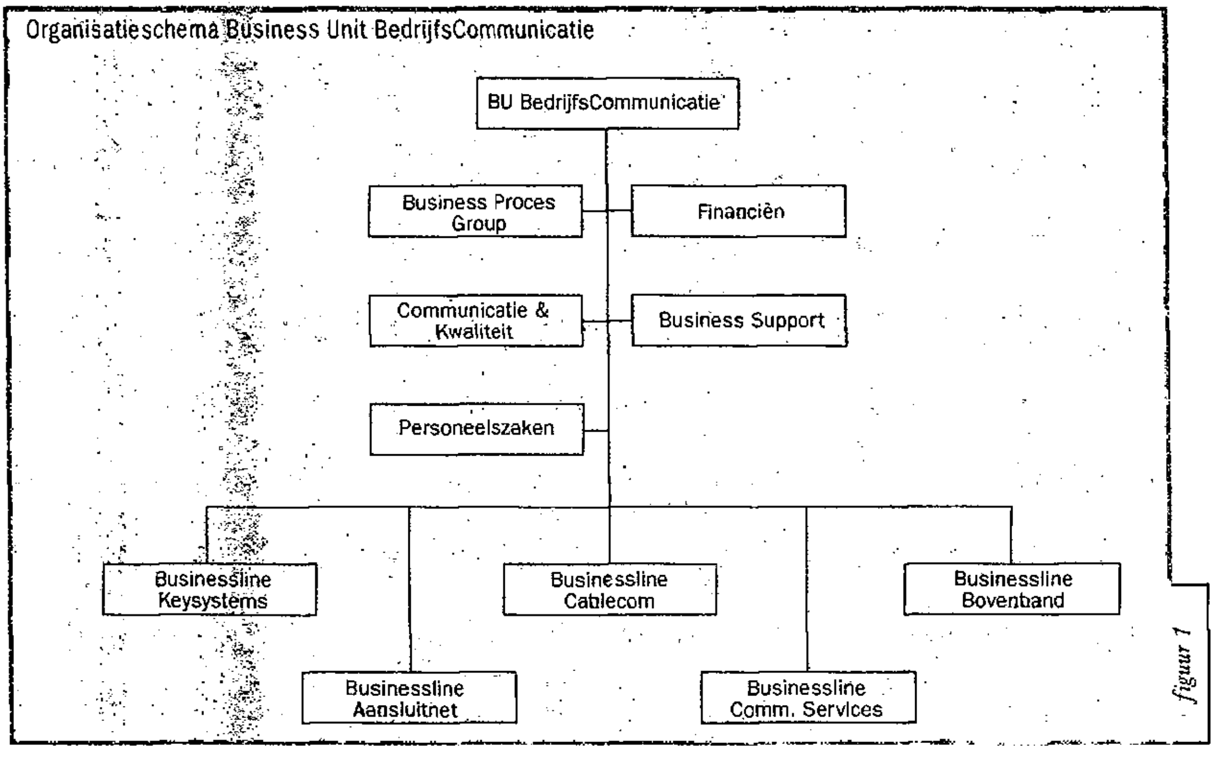
De BU BC bestaat uit de volgende bedrijfsonderdelen (zie figuur 1):
- Business Line Bovenband;
- Business Line Keysystems;
- Business Line Cablecom;
- Business Line Aansluitnet;
- Business Line Communication Services;
- Business Proces Group;
- Communicatie en Kwaliteit;
- Business Support;
- Personeelszaken;
- Financiën.

2 Probleemstelling

2.1 Inleiding

In hoofdstuk 1 is aangegeven dat noch eilandautomatisering noch één allesomvattend informatiesysteem op adequate wijze de bedrijfsprocessen van een organisatie kunnen ondersteunen. En daar gaat het toch om, dat is de toegevoegde waarde van ICT.

De praktijkervaringen van de auteurs van dit artikel sluiten nauw aan bij de visie van prof. Nielen [Niel1989]. Deze stelt onder andere:
- grote informatiesystemen moeten niet als één geheel worden ontworpen;
- als we de omvang van een informatiesysteem vergroten neemt de moeilijkheidsgraad exponentieel toe;
- het is verstandig om verschillende systemen te hanteren, die ieder een specifiek stel eigenschappen hebben;
- grote informatiesystemen mogen niet meer lijken op een kantoorgebouw dat ontworpen, gebouwd en betrokken wordt; ze moeten daarentegen lijken op het personeel van de organisatie dat verloopt en wordt aangevuld;
- voor het onderkennen van de optimale subsystemen is



figuur 1

een grondige kennis vereist van de onderneming waarvan het informatiesysteem de besturing ondersteunt;
- samenvattend: informatiesystemen moeten niet te groot zijn, niet teveel functionaliteiten bezitten en moeten weggegooid en vervangen kunnen worden wanneer ze niet meer voldoen.

In 1998 heeft de BU BC, geconfronteerd met haar verzameling informatiesystemen en interfaces, geconcludeerd (probleemstellingen) dat:
- sommige informatiesystemen teveel functionaliteiten bezitten;
- sommige informatiesystemen een monolithisch karakter hebben;
- er erg veel 1 op 1 interfaces zijn;
- het daardoor moeilijk wordt om informatiesystemen te vervangen;
- het moeilijk is om snel in te spelen op veranderingen in de business;
- het onderhoud van de informatiesystemen en interfaces zeer hoge kosten met zich meebrengt.

Dat heeft geleid tot een aantal belangrijke beslissingen (doelstellingen):
- informatiesystemen voor de BU BC moeten worden gebouwd volgens de drie-lagen-architectuur: datalaag, functielaag en presentatielaag [Stee1997];
- per laag worden zogenoemde bouwblokken gemaakt, die door meerdere informatiesystemen kunnen worden gebruikt;
- het bouwen van nieuwe informatiesystemen moet daardoor sneller en goedkoper;
- hergebruik van componenten wordt gestimuleerd (beloond);
- een uitdaging is het realiseren van een adequate verzameling bouwblokken.
Daartoe zou gebruik kunnen worden gemaakt van het "Biljartmodel"(zie figuur 2). Een beginnende biljarter is blij als hij de stootbal goed raakt en op die manier een carambole maakt. Een professionele biljarter wil niet alleen de carambole maken, maar ook iets goeds (een goede ligging van de ballen) overhouden. Van een professionele informaticus mag men iets dergelijks verwachten. Een professionele projectleider scoort niet alleen (opleveren van afgesproken functionaliteiten, op tijd, binnen het budget, voldoen aan kwaliteitseisen) maar houdt ook iets goeds (bouwblokken waar de organisatie iets aan heeft) over. Dit is een interessant beoordelingsaspect;
- de verzameling interfaces moet worden gesaneerd. Dit onderwerp staat centraal in dit artikel.

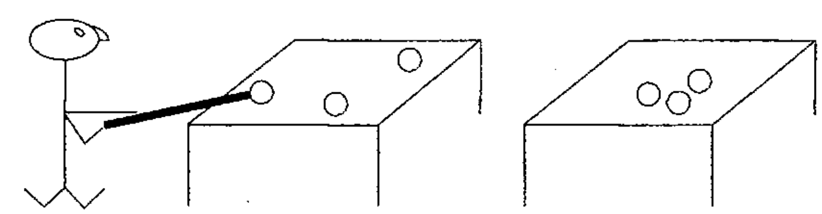
2.2 Probleem

Gegevensuitwisseling tussen informatiesystemen kan op verschillende manieren.

Vanouds is er de 1 op 1 koppeling/interface. Figuur 3 schetst zo'n interface tussen twee informatiesystemen van de Technische Universiteit Delft die op die manier hebben gefunctioneerd in de periode 1985-1999 [Dijk84]. Het voorraadbeheer- en magazijnsysteem GEMS ontving iedere nacht per magazijn de geldige klantnummers. Het financieel systeem FINS ontving periodiek de factureringsgegevens. Deze 1 op 1 interface (in feite zijn het twee interfaces) is gemaakt in 1985. Hij beviel zo goed, dat soortgelijke interfaces zijn gemaakt voor FINS met andere informatiesystemen. Daardoor ontstond een 1 op n interface (zie figuur 4). Al filosoferend kwam de wens op tafel om te komen tot een algemeen interfacemechanisme voor n op m koppelingen (zie figuur 5). Een dergelijke, algemeen toepasbare, oplossing viel echter buiten de scope van de TU Delft.

De huidige message brokers bieden dergelijke faciliteiten. Dit was voor de Business Proces Group aanleiding om in 1998 tot aanschaf van de message broker ENGIN

Biljartmodel (raken en iets goeds overhouden)



figuur 2

DE MENING VAN ... MONIQUE CLAUS

Monique Claus is werkzaam bij de Business Proces Group van de Business Unit BedrijfsCommunicatie. Zij is betrokken bij architectuurvraagstukken.
Monique: "Wij richten ons bij applicatieontwikkeling sterk op de drie-lagen-architectuur: presentatielaag, functielaag en gegevenslaag. Nieuwe applicaties, zoals CAPS dat wordt gebruikt voor de afhandeling van capaciteitsaanvragen, worden volgens deze architectuur opgezet. Bestaande applicaties worden bij groot onderhoud "uit elkaar getrokken".
Naast deze drie-lagen-architectuur werken wij steeds meer met componenten of bouwblokken. Deze kunnen door meerdere applicaties worden gebruikt. Op die manier bestaat een applicatie voor een belangrijk deel uit een verzameling bouwblokken met een besturingselement. De vraag is hoe bouwblokken met elkaar communiceren. Soms communiceren de bouwblokken via de message broker. Dat werkt prima. Als de bouwblokken klein zijn, worden ze vaak geassembleerd tot een groter bouwblok. Als we alle kleine bouwblokjes via de message broker zouden laten communiceren, dan zou dat erg veel "overhead" geven en in sommige gevallen vertragend werken. Toch is de omvang van een bouwblok niet altijd bepalend. Zo hebben we een bouwblok voor assortimentsgegevens dat niet zo groot is, maar dat wel door allerlei applicaties wordt geraadpleegd. De message broker speelt daarbij een essentiële rol.

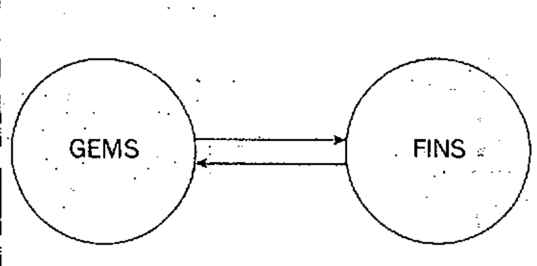
De keuze of bouwblokken worden samengevoegd tot een groter bouwblok of dat ze communiceren via de message broker moet iedere keer zorgvuldig worden afgewogen!".

over te gaan.

De probleemstelling:

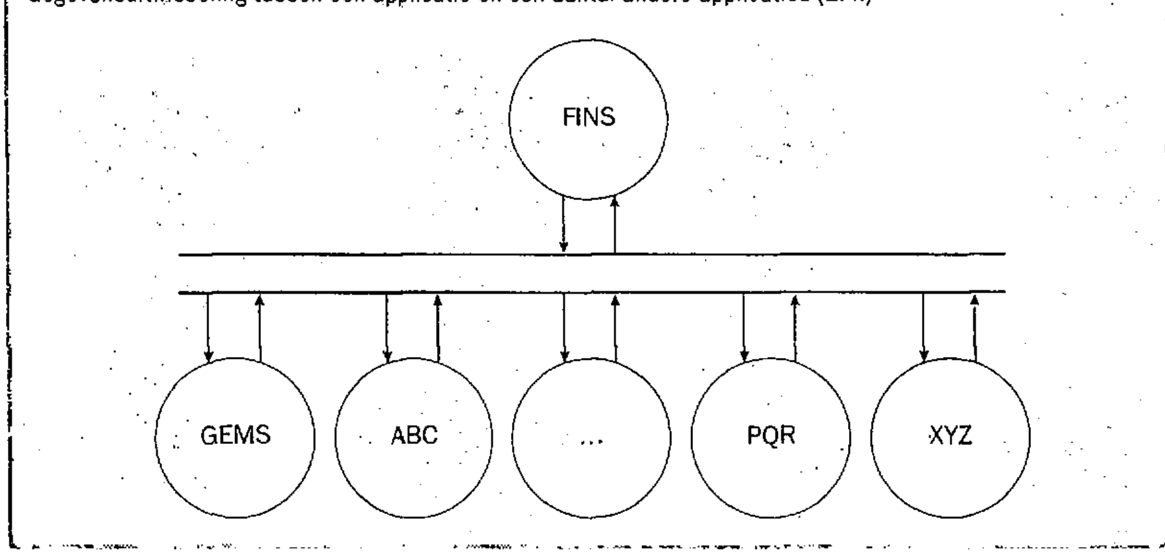
"Er zijn teveel 1 op 1 interfaces en het onderhoud van de interfaces brengt te hoge kosten met zich mee" kon als volgt worden vertaald naar een oplossing:
- realiseer zoveel mogelijk nieuwe interfaces via de message broker;
- vervang bij groot onderhoud van de huidige interfaces deze door interfaces via de message broker, tenzij er door het management goed te keuren redenen zijn om

Gegevensuitwisseling tussen twee applicaties (1:1)



figuur 3

Gegevensuitwisseling tussen een applicatie en een aantal andere applicaties (1: n)



figuur 4

dat niet te doen;
- rapporteer over de voor- en nadelen van het gebruik van de message broker.

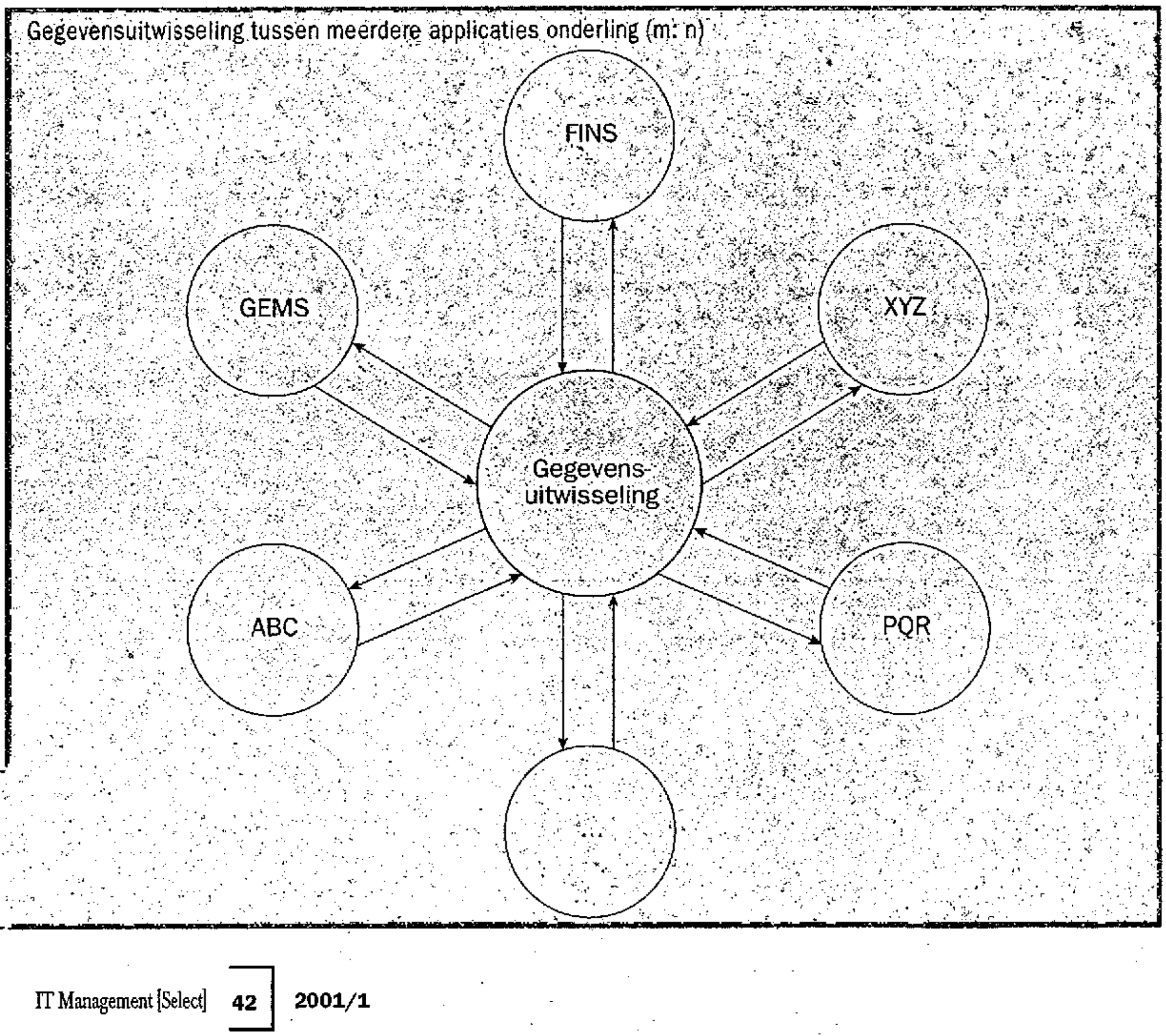
3 Casus

3.1 Inleiding

Zoals in de vorige hoofdstukken is aangegeven, heeft de BU BC een groot aantal informatiesystemen, waaronder een aantal "landelijke systemen" [Clau1999]. Deze informatiesystemen hebben vaak onderlinge koppelingen/interfaces ten behoeve van gegevensuitwisseling. In dit hoofdstuk worden in het kort twee, voor de BU BC zeer belangrijke, informatiesystemen alsmede een interface tussen deze informatiesystemen besproken. De betreffende informatiesystemen en de interface komen ook in het volgende hoofdstuk aan de orde.

3.2 M&M007

Het informatiesysteem M&M007 (Meld- en Meetpost 007) wordt gebruikt voor de registratie en afhandeling van storingsmeldingen betreffende telecommunicatieapparatuur en – installaties. Met behulp van M&M007 worden storingsmeldingen geregistreerd, geanalyseerd en afgehandeld. Het informatiesysteem biedt ondersteuning bij het vertalen van de storing naar een serviceorder en gebruikt, indien beschikbaar, daarbij het betreffende servicecontract.



figuur 5

Het informatiesysteem ondersteunt onder andere de volgende processen:
- het aannemen van klachtmeldingen;
- het analyseren van klachten;
- het verstrekken van serviceorders aan monteurs om een storing op te heffen;
- bewaken van de voortgang van de afhandeling van een klachtmelding;
- verstrekken van informatie over de klachtmeldingen;
- gereed melden aan de klant van de klachtafhandeling.

M&M007 heeft interfaces met acht informatiesystemen, waaronder FIT.

3.3 FIT

Met behulp van het informatiesysteem FIT (Field Information Terminal) worden monteurs elektronisch aangestuurd. Daartoe beschikken de monteurs over een mobiele werkplek en een GSM toestel. FIT bestaat uit twee deelsystemen: FIT Centraal en FIT Mobiel. FIT Centraal wordt gebruikt voor toewijzing, planning, bewaking van klantorders en het vervaardigen van managementinformatie. FIT Mobiel is het deel van het FIT informatiesysteem dat op de PC's van de monteurs is geïnstalleerd. De monteurs gebruiken FIT Mobiel voor de afhandeling van klantorders. Het informatiesysteem ondersteunt onder andere de volgende processen:
- in behandeling nemen van serviceorders;
- toewijzen van serviceorders;
- bewaken van de voortgang van serviceorders;
- verwerken van materiaalaanvragen;
- bestellen van artikelen;
- afmelden uitgevoerde werkzaamheden bij de klant;
- gereed melden van de serviceorder.

FIT heeft interfaces met zes informatiesystemen, waaronder M&M007.

3.4 Interfaces M&M007 en FIT

Tussen M&M007 en FIT bestaan de volgende interfaces:
- M&M007 biedt serviceorders aan aan FIT die deze in ontvangst neemt;
- FIT informeert M&M007 over de voortgang van de serviceorders en biedt terug- en gereedmeldingen aan.

Tabel 1 geeft een overzicht van de betreffende transacties.

4 Koppeling van applicaties

4.1 De rol van de message broker

Enterprise Application Integration (EAI) is het proces dat de machines van een of meer bedrijven via een netwerk met elkaar verbindt [Rock2000], zodat zowel nieuwe applicaties als legacy-applicaties en hun gegevensbestanden kunnen worden geïntegreerd. De integratie vereist daarbij weinig of geen wijzigingen van de bestaande applicaties. Een belangrijke EAI-component is de message broker. Gartner [Dec1998] geeft als definitie en doel van een message broker:

"Message brokers are logical hubs that copy and forward messages to one or more destinations. A message broker is an intelligent third party (hence a 'broker') between information sources and information consumers. The purpose of this architectural type is to integrate independently designed application domains (e.g., new applications, purchased applications and legacy applications). Message brokers make it possible to re-engineer business processes without re-engineering all the application programs and databases."

Een message broker is een (centraal gepositioneerde) middleware applicatie die het mogelijk maakt om andere applicaties eenvoudiger met elkaar te laten communiceren door het uitwisselen van gegevens. Er kunnen drie vormen van gegevensuitwisseling worden onderscheiden [Over1998], te weten:
- *Een vraag om gegevens*: applicatie A stelt een vraag aan applicatie B. Deze stuurt het antwoord op de vraag of een afwijzing (voorbeeld: "niet geautoriseerd voor deze gegevens");
- *Een verzoek tot wijziging van gegevens*: applicatie A wil dat applicatie B een of meer gegevens wijzigt. Dit kan een inhoudelijke wijziging zijn (voorbeeld: de mededeling dat een adres is gewijzigd) of het toevoegen of verwijderen van een gegeven (voorbeeld: toevoegen van een adres);
- *Een opdracht tot wijziging*: applicatie A meldt aan applicatie B dat er iets van zijn gegevens moet worden

Tabel 1: Overzicht interfaces M&M007 waaronder die met FIT

M&M007 transacties		**FIT transacties**
Serviceorder verwerken door externe systemen:		
Leveren van een serviceorder	ext-ord-1	Ontvangen van een serviceorder
Ontvangen van een terugmelding	ext-ord-2	Leveren van een terugmelding
Ontvangen van voortgangsinformatie	ext-ord-3	Leveren van voortgangsinformatie
Ontvangen van een gereedmelding	ext-ord-4	Leveren van een gereedmelding
Serviceorder verwerken door M&M007:		
Ontvangen van een serviceorder	int-ord-1	
Leveren van voortgangsinformatie	int-ord-2	
Leveren van een gereedmelding	int-ord-3	
Leveren van informatie over de serviceorder	int-ord-4	

gewijzigd (voorbeeld: het wijzigen van een projectstatus). Deze laatste vorm doet zich voor wanneer applicatie A de "eigenaar" is van bij applicatie B opgeslagen gegevens.

Applicaties hebben vaak een verschillend datamodel. Daarom is het wenselijk om tijdens het uitwisselen van gegevens de mogelijkheid te hebben om gegevens te transformeren. Onder transformeren wordt in dit geval zowel *converteren* (veldinhoudelijke wijzigingen, bijvoorbeeld het omzetten van een datum) als *mappen* (het omzetten van recordlayouts) verstaan. Daarmee is de communicatie op applicatieniveau geregeld. Naast deze communicatielaag is er de communicatie op transportniveau. Daarin wordt beschreven hoe de zender en ontvanger samenwerken. De twee lagen van communicatie zijn in principe onafhankelijk van elkaar.

Samenvattend kan worden gesteld dat een message broker een snelle postbode is (gegevensuitwisseling) die, alvorens de gegevens te bezorgen, de gegevens op een door de ontvanger gewenste manier omzet (transformeren) en daarmee een uitstekend communicatiemiddel is om applicaties met elkaar te laten praten.

De BU BC heeft in 1998 de message broker ENGIN van de Amerikaanse firma Muscato aangeschaft. Het uitwisselen en transformeren kan worden toegelicht aan de hand van figuur 6.

Iedere aan de message broker aangesloten applicatie verleent een aantal diensten met betrekking tot gegevensuitwisseling. Daartoe wordt van iedere dienst een zogenoemde Service SPecificatie (SSP) gemaakt. Dus applicatie A zegt als het ware: hier is mijn dienst A3 en wie de dienst wil en mag afnemen, vraagt dat maar aan de message broker. Het zal applicatie A, tot op zekere hoogte, een zorg zijn welke applicaties allemaal dienst A3 afnemen. Indien nu applicatie A volgens dienst A3 (voorbeeld: verzenden van projectgegevens) gegevens wil verzenden naar applicatie B met dienst B5 (voorbeeld: ontvangen van projectgegevens), dan zal de message broker de betreffende gegevens van applicatie A naar applicatie B sturen. Alvorens de gegevens aan applicatie B aan te bieden zal de message broker de gegevens transformeren. De wijze waarop dit dient te geschieden is binnen de message broker vastgelegd met behulp van een Service Implementatie SPecificatie (SISP).

Zonder te streven naar volledigheid wordt nu op een aantal aspecten van het werken met een message broker nader ingegaan.

Interface-awareness

De message broker kan onder andere worden gebruikt om gegevensuitwisseling tussen bestaande legacy-systemen te realiseren. Dat neemt niet weg dat moet worden gestreefd naar "interface-awareness" van applicaties. Dat betekent dat applicaties er rekening mee houden dat ook de buitenwereld de gegevens en de functionaliteit van de applicatie wil gebruiken. Omgekeerd moeten applicaties er rekening mee houden dat sommige gegevens uit de buitenwereld komen. Door een goede "Interface-awareness" van de op de message broker aangesloten applicaties wordt niet alleen de rol van de message broker beperkt, maar kan ook sneller worden ingespeeld op de veranderende informatiebehoefte van de organisatie. De kans dat een legacy-systeem opgebouwd is volgens het drie-lagen-model is gering. Presentatielaag, functielaag en datalaag zijn vaak vergaand met elkaar verweven. Om de functies van een legacy-systeem beschikbaar te stellen aan de buitenwereld kan (een deel van) de functielaag worden gesimuleerd met behulp van zogenoemde adapters. In paragraaf 4.2 komt een en ander aan de orde. Als applicaties gegevens nodig hebben van of moeten leveren aan andere applicaties neemt de onderlinge afhankelijkheid van de applicaties toe. Interface-awareness betekent ook dat applicaties niet mogen "stuklopen" omdat een andere applicatie tijdelijk niet beschikbaar is.

Domeinen

Als het aantal applicaties dat via de message broker wil communiceren erg groot wordt of als er bijvoorbeeld meerdere opdrachtgevers (met wellicht tegengestelde belangen) zijn, dan is het vanuit het oogpunt van beheer verstandig om meerdere (kopieën van) message brokers te installeren. Elke message broker krijgt dan zijn eigen domein waartoe een aantal applicaties behoort. Applicaties die sterk van elkaar afhankelijk zijn, moeten bij voorkeur tot hetzelfde domein behoren. Daardoor kan de communicatie tussen domeinen beperkt blijven. In paragraaf 5.2 wordt aangegeven dat de message broker ENGIN van de BU BC als domein gaat functioneren binnen het grotere geheel van KPN Telecom.

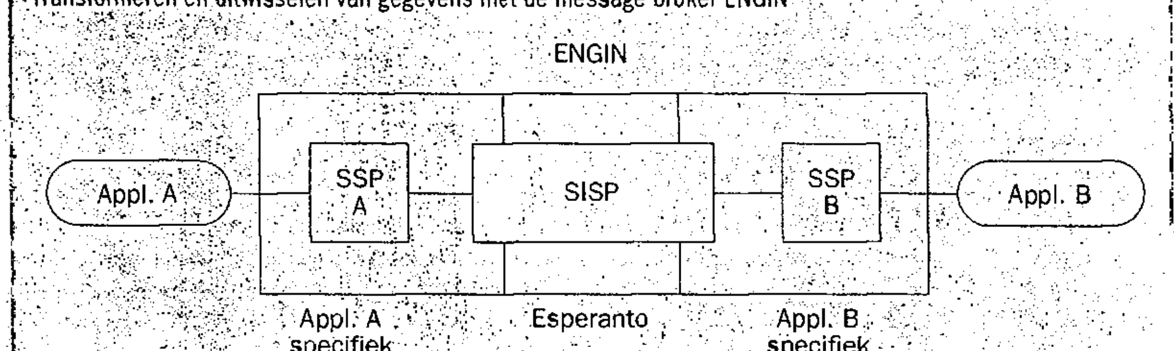
Single point of failure

Het gebruik van een message broker voor het uitwisselen en transformeren van gegevens tussen applicaties is, ook uit oogpunt van beheer, een zeer verstandige zaak. Maar zoals zo vaak, kleven er aan de invoering van een message broker ook nadelen. Doordat de gegevensuitwisseling van veel applicaties via de message broker loopt, is er als het ware een "single point of failure" in de infrastructuur ontstaan. Als de message broker onderuit gaat, dan ligt alle communicatie tussen de applicaties stil. De risico's van een single point of failure kunnen tegenwoordig worden ingeperkt door het nemen van adequate (tegen)maatregelen. In dit artikel wordt hierop niet verder ingegaan.

Esperanto

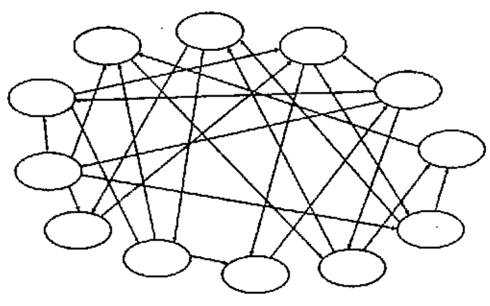
Zoals in de probleemstelling is aangegeven heeft de BU BC teveel 1 op 1 interfaces (zie figuur 7) en brengt het onderhoud te hoge kosten met zich mee. Door interfaces

Transformeren en uitwisselen van gegevens met de message broker ENGIN



figuur 6

Het beeld van een groot aantal 1 op 1 interfaces



figuur 7

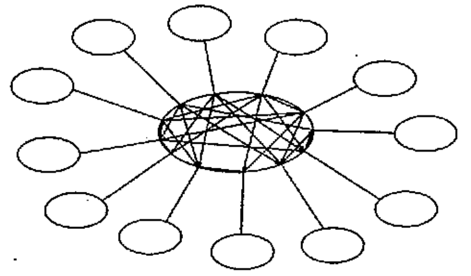
aan te sluiten op een message broker blijken de kosten sterk teruggebracht te kunnen worden, onder andere omdat:

- de interfaces kunnen worden gestandaardiseerd;
- hergebruik van oplossingen mogelijk is;
- kennis van het gebruik van de message broker gebundeld is, dit in tegenstelling tot de grote verzameling 1 op 1 interfaces waarvan doorgaans de bouwers niet meer beschikbaar zijn en het beheer niet altijd adequaat is ingericht;
- de message broker al veel werk kan verrichten tijdens het transformeren;
- beheer centraal kan plaatsvinden.

Maar hoe zit het met het aantal interfaces?

In principe verandert het aantal interfaces niet (zie figuur 8). Van iedere interface moet een SISP (transformatieroutine) gemaakt worden. Dit is in het algemeen een stuk eenvoudiger dan het vervaardigen van een 1 op 1 interface. Toch wil de BU BC naar minder interfaces toe. Daartoe heeft het Software Huis binnen de message broker een soort "Esperanto" (tussentaal) gedefinieerd. Alle data wordt eerst vertaald naar de Esperanto-vorm en vervolgens naar het gewenste "dialect"van de bestemmingsapplicatie. Dit betekent dat binnen de message broker een uniform datamodel wordt gehanteerd. Alle data worden daarbij tweemaal getransformeerd. Het aantal transformatieroutines kan daardoor worden beperkt tot 2*n (voor iedere applicatie zijn transformatieroutines van en naar het uniform datamodel nodig), als n het aantal aangesloten applicaties is. Dat is aanzienlijk minder (zie figuur 9) dan in het geval van directe transformaties waarin, net als bij 1 op 1 koppelingen, het maximum aantal transformatieroutines n*(n-1) bedraagt. Ook het onderhoud is minder arbeidsintensief. Indien van applicatie A, die met m andere applicaties een koppeling heeft, het interface-record wijzigt, moeten bij directe transformatie 2*m transformatieroutines worden aangepast terwijl in het geval van het gebruik van Esperanto slechts 2 transformatieroutines behoeven te worden aangepast. Alleen in het geval dat de wijzigingen dusdanig zijn dat het Esperanto, het uniforme datamodel, voor de betrokken applicaties moet worden aangepast moeten 2*m transformatieroutines worden aangepast. Dit laatste zal, indien het uniforme datamodel zorgvuldig is opgesteld, slechts in zeer beperkte mate gebeuren. In tabel 2 is een overzicht van een en ander gegeven.

Een groot aantal interfaces dat werkt via de message broker



figuur 8

Transport

Om berichten te kunnen versturen, moeten ze worden beschreven of populair gezegd "ingepakt". De ontvanger van een bericht kan het weer "uitpakken" als hij weet hoe de verpakking eruitziet. De huidige versie van ENGIN werkt met segmenten als beschrijvingsmethode. Ook ander standaarden (bijvoorbeeld EDI en HL7) werken op basis van segmenten. Ondersteuning door ENGIN van XML (eXtensible Markup Language) is sinds kort mogelijk.

4.2 De praktijk

De message broker is sinds 1998 bij de BU BC in gebruik. Hoe zijn de ervaringen? De BU BC heeft in nauwe samenwerking met het Software Huis gekozen voor een kwalitatieve benadering en is zeer zorgvuldig te werk gegaan. Stap voor stap worden interfaces aan de message broker gekoppeld. De opgedane ervaringen worden gebruikt om tot een proces van continue kwali-

Tabel 2: Benodigd aantal transformatieroutines bij n applicaties

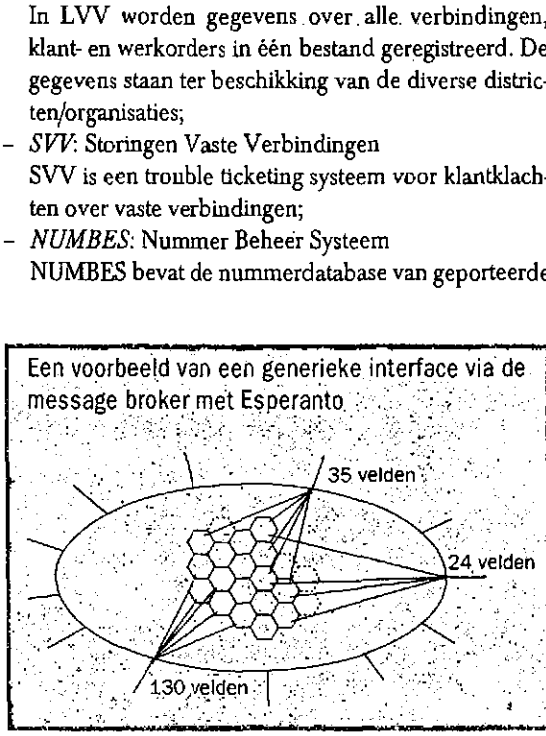
Benodigd aantal transformatieroutines bij n applicaties			
	directe trans-formatie	*gebruik van een Esperanto*	*conventionele architectuur (1 op 1 kopp.)*
max. aantal routines	n*(n-1)	2*n	n*(n-1)
min. aantal routines te wijzigen	2*m	2	2*m
max. aantal routines te wijzigen	2*m	2*m	2*m
n = het aantal applicaties m = het aantal betrokken applicaties bij wijziging van een interface-record			

teitsverbetering te komen. In de literatuur staat dit bekend als de cirkel van Deming ("Plan-Do-Check-Act"-cirkel) [Dorr1999]. Voor het aansluiten van de eerste interfaces is ruimschoots de tijd genomen. De snelheid van aansluiten neemt echter drastisch toe. Het voorbereiden, de functionele fase, kost doorgaans vrij veel tijd. Indien de functionaliteit van een interface goed gedefinieerd is, dan is de technische realisatie binnen de message broker doorgaans binnen enkele dagen of weken geregeld. Verwacht wordt dat het aantal koppelingen in het jaar 2001 sterk toeneemt.

Om inzicht te geven in het soort applicaties dat thans gekoppeld is aan de message broker wordt hierna een aantal applicaties genoemd dat is aangesloten op de message broker. In paragraaf 4.2.1 wordt beschreven hoe een interface van M&M007 met FIT functioneert. In paragraaf 4.2.2 wordt beschreven hoe onlangs een interface tussen M&M007 en Baan ERP is gerealiseerd. Voorbeelden van applicaties die interfacen via de message broker zijn:

- *M&M007*: Meld- en Meetpost 007
  Registratie en afhandeling van storingsmeldingen betreffende telecommunicatieapparatuur en -installaties;
- *FIT*: Field Information Terminal
  Met behulp van FIT worden monteurs elektronisch aangestuurd. Daartoe beschikken de monteurs over een mobiele werkplek en een GSM toestel (de zogenoemde "FIT-koffer");
- *CKR*: Centrale Klanten Registratie
  CKR ondersteunt het klantgericht werken door een eenduidige identificatie van klanten mogelijk te maken en inzicht te bieden in de relatie tussen een klant en KPN Telecom;
- *SIZM*: Service Informatie Zakelijke Markt
  SIZM bevat de prijzen van service-artikelen. Deze prijzen worden gebruikt in het servicewerkveld voor het doorberekenen van verleende services en verbruikte materialen;
- *CAPS*: Capaciteits Planning Systeem
  Met behulp van CAPS wordt de beschikbare monteurscapaciteit beschikbaar gesteld aan de verkoopeenheden;
- *KIOSK*: Klant Interactie Ondersteunend Systeem
  KIOSK ondersteunt de eenheid "Sales Bedrijven" bij de verkoop, levering en facturering van standaardorders. In KIOSK worden gegevens betreffende orders, assortiment en klanten vastgelegd;
- *LVV*: Leveringssysteem Vaste Verbindingen
  In LVV worden gegevens over alle verbindingen, klant- en werkorders in één bestand geregistreerd. De gegevens staan ter beschikking van de diverse districten/organisaties;
- *SVV*: Storingen Vaste Verbindingen
  SVV is een trouble ticketing systeem voor klantklachten over vaste verbindingen;
- *NUMBES*: Nummer Beheer Systeem
  NUMBES bevat de nummerdatabase van geporteerde

Een voorbeeld van een generieke interface via de message broker met Esperanto



figuur 9

telefoonnummers met daarbij de operator waarnaar het telefoonnummer is geporteerd;
- *AWO*: Automatisering Werk Orders
  AWO ondersteunt het "consumentenproces" bij de verkoop, levering en facturering van standaardorders.

4.2.1 Koppeling van serviceorders en monteurs

Zoals in hoofdstuk 3 is vermeld, is M&M007 het informatiesysteem dat wordt gebruikt voor de registratie en afhandeling van storingsmeldingen betreffende telecommunicatieapparatuur en -installaties. M&M007 biedt ondersteuning bij het vertalen van een storing naar een serviceorder. M&M007 kan worden gebruikt om serviceorders af te handelen, maar het is ook mogelijk om serviceorders (door) te leveren aan een ander informatiesysteem, dat de betreffende serviceorders afhandelt en soms ook weer terugstuurt. Het FIT informatiesysteem ontvangt van M&M007 serviceorders ter afhandeling, door en ter ondersteuning van de monteurs, en ook Baan ERP ontvangt van M&M007 serviceorders. Om een en ander te regelen heeft M&M007 een "serviceorder interface" die een aantal transacties (zie tabel 1) omvat. De service/dienst die M&M007 met deze interface verleent, wordt beschreven met behulp van een Service SPecificatie (SSP). Ook de service die FIT verleent, wordt in een SSP beschreven. Het transformeren gebeurt door ENGIN op basis van de Service Implementatie SPecificatie (SISP). In het kort wordt nu ingegaan op de SSP en SISP.

Inhoud van het SSP "M&M007 serviceorder interface"

De volgende onderwerpen maken deel uit van het SSP:
- de betreffende transacties;
- validaties door de message broker ENGIN;
- validaties door M&M007;
- acties;
- foutafhandeling door het externe systeem;
- berichtenspecificatie.

Transacties

Transacties om serviceorders te laten verwerken door externe systemen:
- ext-ord-1: Leveren van een serviceorder;
- ext-ord-2: Ontvangen van een terugmelding;
- ext-ord-3: Ontvangen van voortgangsinformatie;
- ext-ord-4: Ontvangen van een gereedmelding.

Transacties om serviceorders (intern) te verwerken door M&M007:
- int-ord-1: Ontvangen van een serviceorder;
- int-ord-2: Leveren van voortgangsinformatie;
- int-ord-3: Leveren van een gereedmelding;
- int-ord-4: Leveren van informatie over de serviceorder.

Met behulp van de transactie ext-ord-1 meldt M&M007 een serviceorder aan bij een extern informatiesysteem, bijvoorbeeld FIT. Het betreffende externe informatiesysteem blijft eigenaar van de serviceorder totdat het een terug- of gereedmelding aan M&M007 stuurt. Het externe informatiesysteem kan gedurende de afhandeling van de serviceorder rapporteren over de voortgang.

Validaties

Door de message broker ENGIN wordt een beperkt aantal validaties uitgevoerd. De belangrijkste validatie is: "Het bericht moet voldoen aan het in ENGIN gespecificeerde formaat". M&M007 stelt duidelijke eisen aan het "Ontvangen van een serviceorder" (transactie int-ord-1). Tabel 3 geeft enkele validaties weer.

Acties/Foutafhandeling

Als alle validaties tot een goed resultaat leiden voert M&M007 een aantal activiteiten uit om de serviceorder op te slaan. Het externe informatiesysteem kan in het veld "resultaatcode" zien of de serviceorder succesvol is opgeslagen of dat er fouten zijn opgetreden.

Berichtenspecificatie

In de berichtenspecificatie wordt aangegeven hoe het interface-record van de "M&M007 serviceorder interface" eruitziet.

Inhoud van het SISP "M&M007 – FIT serviceorder interface"

In de SISP worden de mappings, waarmee records van M&M007 worden afgebeeld op records van FIT vastgelegd. Tabel 4 bevat enkele velden van de mapping van de transactie "Leveren van een serviceorder" door M&M007 aan FIT. Het werkelijk aantal velden bedraagt 130 (zie figuur 9).

Tabel 3: Enkele validaties van de transactie "Ontvangen van een serviceorder"

Validatie	Foutcode
Het externe informatiesysteem moet in M&M007 geautoriseerd zijn voor deze transactie	100
De in het bericht als verplicht gemarkeerde velden moeten gevuld zijn	200
Als "contractsoort" gevuld is, dan moet "berekend tijdstip" ook ingevuld zijn en vice versa	206
De "code productsoort" moet bestaan in het gegevensbouwblok "assortiment"	400

Tabel 4: Enkele velden van de mapping van de transactie "Leveren van een serviceorder" door M&M007 aan FIT

Veld van M&M007	Veld van FIT
MELD_S.klantorder nummer	KLA_ID
MELD_S.telefoonnummer	KLA_TCNR
MELD_S.product-ID	KLA_LIJNBENA
VERPL_S.urgentie code	URG_CODE
WERK_S.uitgifte datum	KLA_DT_UITG
KLANT_S.klantnaam	KLA_NAAM
WERK_S.analist code	ANA_CODE
MELD_S.order type	KLA_INFO

DE MENING VAN ... XANORA KOOL

Xandra Kool was tot voor kort werkzaam bij het "Coördinatiecentrum services" van de Productline Ericsson (PLE) van de BU BC. Zij was als keyuser intensief betrokken bij het pilotproject PERFECTO dat tot doel had het "vererpen" van de informatievoorziening binnen de PLE.

Een essentieel onderdeel van PERFECTO was het realiseren van koppelingen van Baan ERP met bestaande informatiesystemen van PLE. Om een tussentijdse beoordeling van het project mogelijk te maken wenste de stuurgroep op een bepaald moment een demonstratie, inclusief de interfacemogelijkheden.

De functionele specificaties van de interfaces waren op dat moment reeds opgesteld. Binnen twee weken werkten de interfaces met de informatiesystemen M&M007 (call registratie) en FIT (opdrachten voor de Field Engineers), die met Baan ERP het proces "Uitvoeren correctief onderhoud n.a.v. een call" ondersteunen.

Xandra: "De snelheid waarmee een en ander is gerealiseerd heeft mij aangenaam verrast. Dit was mede mogelijk omdat de applicaties M&M007 en FIT reeds door middel van een generieke interface met de message broker waren verbonden. Daardoor behoefden deze applicaties niet te worden gewijzigd. Wijzigen van deze applicaties zou, in verband met de, zeer zorgvuldig werkende, change procedure een aanzienlijk langere doorlooptijd hebben betekend. Naar mijn mening is dit een duidelijke verbetering van de slagkracht van de informatievoorziening.
Toch heb ik nog wel wat op te merken. Twee andere applicaties waren via een 1-op-1 verbinding met elkaar gekoppeld. Soms ging er iets fout. Aangezien ik de applicaties en de technische omgeving door en door kende, had ik wat trucjes achter de hand om de productie toch doorgang te kunnen laten vinden.
Sinds enige tijd zijn de betreffende applicaties gekoppeld via de message broker. Een enkele keer gaat er iets fout met de verbinding. In die situaties moet ik nu contact opnemen met "interfacebeheer" die het probleem verhelpt. Vanuit KPN bekeken is het natuurlijk beter dat het "interfacebeheer" is geregeld en dat de technische kennis van de interfaces geconcentreerd is in een team rondom de message broker. Maar ik vind het wel jammer van mijn trukendoos!"

4.2.2 Koppeling van serviceorders met Baan ERP

Ook Baan ERP ontvangt serviceorders van M&M007. Daartoe is een SSP voor Baan ERP gemaakt en een SISP voor de mapping van het M&M007 serviceorder-record op het betreffende Baan ERP-record. Een groot voordeel was dat aan de kant van M&M007 geen wijzigingen nodig waren. Wijzigingen aanbrengen in een dergelijke cruciale applicatie vergt doorgaans een aanzienlijke doorlooptijd, die door de gekozen aanpak, een *generieke interface* met SSP, kan worden voorkomen. Daardoor was het mogelijk de gewenste koppeling in korte tijd te realiseren. De benodigde tijd ging vooral zitten in het aansluiten van de nieuwe applicatie Baan ERP.

ERP pakketten staan niet bekend om hun openheid met betrekking tot het communiceren met andere applicaties. Baan Company heeft met versie V, ook wel bekend onder de naam Correlli, een substantiële stap in de richting van openheid gedaan [Baan1999]. Met versie V is het mogelijk om *on line* te communiceren. Daartoe is er in versie V sprake van een lagenstructuur. Het is nu mogelijk om ook op een andere manier dan via on line Baan-sessies met Baan ERP te communiceren. Daartoe heeft men zogenoemde BOI's gedefinieerd. Een Business Object Interface maakt het mogelijk om op een andere manier dan via de on line Baan-sessies, met behoud van de Baan-functionaliteit, te communiceren met Baan ERP. De snelheid van het realiseren van een koppeling hangt sterk af van de beschikbaarheid van de noodzakelijke BOI. Als deze reeds gebouwd is, kan de koppeling snel worden gerealiseerd. In het andere geval zal de BOI door de leverancier gebouwd moeten worden, hetgeen een zekere doorlooptijd met zich meebrengt. Naast de BOI's is er sprake van DAL's (Dynamic Access Layer). Een DAL fungeert als communicator met de Baan-database (-tabellen). Door deze mogelijkheden kan een externe applicatie communiceren met Baan ERP zonder kennis te hebben van de tabellenstructuur van Baan ERP. Als de ontwikkelafdeling wijzigingen in de tabellenstructuur aanbrengt hoeft dat voor de BOI's en externe applicaties geen gevolgen te hebben.

Adapter

In paragraaf 4.1 is aangegeven dat als een applicatie (een deel) van zijn functielaag beschikbaar wil stellen, dat kan worden gesimuleerd met behulp van adapters. Figuur 6 kan daartoe worden uitgebreid met een adapter. In figuur 10 is dit weergegeven.

Om de ENGIN-Baan adapter te kunnen laten werken zijn ingrediënten (waaronder C-componenten en Java-

DE MENING VAN ... BENNY LANKHORST

Benny Lankhorst is werkzaam bij het "Internet Center" van het Software Huis van KPN Telecom.
Hij heeft in 1999 de opdracht gekregen een web-interface te bouwen voor het informatiesysteem M&M007 (call registratie). Op die manier kunnen "calls" via Internet worden aangeboden aan M&M007.

Benny: "Eindgebruikers zijn in feite niet geïnteresseerd hoe en waarmee een applicatie wordt gebouwd. Wel zijn ze geïnteresseerd in de snelheid waarmee een applicatie beschikbaar komt en de kwaliteit van de applicatie. Alvorens in te gaan op de realisatie kan ik stellen dat het gebruik van de message broker ervoor heeft gezorgd dat de betreffende eindgebruikers de web-interface sneller beschikbaar kregen dan was gepland en eerder dan verwacht. Bovendien hebben de gebruikers opvallend weinig last van storingen. Zij zijn dan ook zeer tevreden.

Als bouwers hadden we een aantal voordelen. We konden ons volledig richten op de web-technologie en hoefden ons niet te bekommeren om de legacy-applicatie die ongewijzigd kon blijven. De moeilijkheidsgraad van de applicatie is daardoor beperkt gebleven. Als twee applicaties interfacen via een 1-op-1 koppeling dan ontstaat er bij een storing vaak een vervelende situatie omdat beide applicatieleveranciers de schuld bij de ander leggen. Nu de applicaties via de message broker interfacen is er als het ware een neutrale component waarmee snel kan worden vastgesteld welke applicatie de storing veroorzaakt.
Ik ben ervan overtuigd dat interfacing met behulp van een message broker niet alleen aanzienlijk voordeliger is dan via traditionele koppelingen, maar dat bovendien de beheerkosten sterk afnemen.

Ik ben voorstander van kleinschalig beginnen en dan stap voor stap uitbouwen in plaats van een werkwijze waarbij een concept dat zich nog niet bewezen heeft in één klap grootschalig wordt geïntroduceerd. Ik voel mij dan ook happy met de gekozen aanpak, met de message broker, door het Software Huis en de BU BC.
Wat ik niet begrijp is dat er zo weinig over dit onderwerp gepubliceerd wordt. Wie weet er nou dat het allemaal zo handig is?".

componenten) van zowel KPN en ENGIN als Baan ERP gebruikt. Figuur 11 geeft dit weer en laat ook de positie van de DAL's zien.

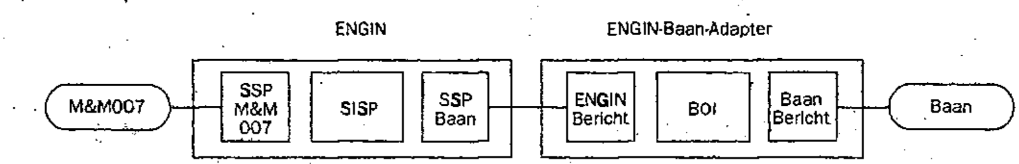
5 Toekomstige ontwikkelingen

5.1 XML

XML, de eXtensible Markup Language is een eenvoudige en zeer flexibele taal die gebaseerd is op SGML (Standard Generalized Markup Language). XML maakt het voor informatieconsumenten en -producenten veel eenvoudiger met elkaar te communiceren. Gegevensuitwisseling tussen applicaties die op verschillende platformen draaien of gebruikmaken van verschillende besturingssystemen kan worden gerealiseerd met XML. XML biedt uitgebreide middelen voor het beschrijven van de inhoud van documenten en van metagegevens. Verwacht wordt dat XML binnen de web-technologie HTML gaat vervangen. De praktijk zal moeten uitwijzen of XML echt een standaard wordt of dat toch weer conversie van de ene XML-omgeving naar de andere XML-omgeving nodig is.

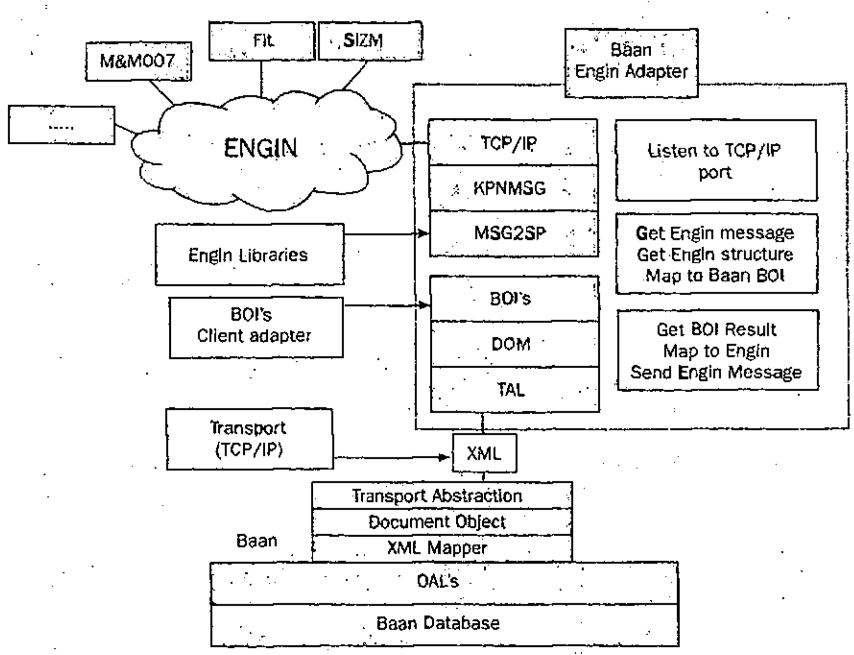
Binnen Baan ERP wordt reeds gewerkt met XML en ook de nieuwe versie van ENGIN ondersteunt XML. Verwacht wordt dat binnenkort binnen BU BC de communicatie via de message broker met behulp van XML plaatsvindt.

ENGIN-Baan adapter



figuur 10

De rol van de BOI's en de DAL's in het communicatieproces



figuur 11

5.2 GAIA

In dit artikel is ingegaan op de casus van de BU BC waarbij de gekozen oplossingen en ervaringen aan de orde zijn gekomen. Parallel aan deze activiteiten wordt binnen KPN Telecom gewerkt aan het project GAIA. GAIA staat voor "Gedistribueerde Applicatie Infrastructuur Architectuur". Dit project wordt op corporate level uitgevoerd. Zodra GAIA operationeel is, sluit de BU BC zich aan bij GAIA en vormt met haar message broker ENGIN een domein binnen GAIA (zie figuur 12). Voor GAIA is het voordeel dat bij de start direct een omvangrijk en goed functionerend domein beschikbaar is.

5.3 Maatschappelijk relevante interfacing

In dit artikel staat het onderwerp interfacing van applicaties centraal. Het gaat daarbij voornamelijk om de interfacing van applicaties binnen een grote organisatie. Een stap verder is het interfacen van applicaties over organisaties heen. In zijn column in IT-Monitor 1/2000 signaleert prof. Nielen [Niel2000] dat na 50 jaar automatisering op basis van een "technology push" heel voorzichtig de "demand pull" boven de horizon komt. "Vragen vanuit de overheid en bedrijfsleven naar oplossingen voor bestaande problemen. Een daarvan betreft het volgende. Informatiesystemen maken gebruik van datadefinities. Wanneer onafhankelijk ontwikkelde informatiesystemen – later – met elkaar moeten communiceren, moet de ene datadefinitie geprojecteerd worden op de andere. Een voorbeeld is het verlangen van de overheid om een aanvraag van sociale bijstand te toetsen aan gegevens van bedrijfsverenigingen en van de belastingdienst. Dergelijke communicatie is vooralsnog razend moeilijk". Nielen is van mening dat een universele interface-generator ontwikkeld moet worden. De auteurs van dit artikel zijn van mening dat de in dit artikel beschreven oplossingen en ervaringen een substantiële bijdrage kunnen leveren om tot een dergelijke interface-generator te komen.

DE MENING VAN … Marc Gofferjé

Marc Gofferjé is manager van de Business Proces Group van de Business Unit BedrijfsCommunicatie.

Marc: "De Business Unit BedrijfsCommunicatie (BU BC) van KPN Telecom opereert zelfstandig als een winstgevende entiteit op de telefoniemarkt. De Business Proces Group (BPG) heeft als kerntaak: het inrichten van, door ICT ondersteunde, processen ten behoeve van de BU. De BU werkt met smalle marges en moet daarom 'op de centen passen'. Daarom is het uit financiële overwegingen niet altijd haalbaar om alle KPN-ICT-standaarden te gebruiken en is de BU voortdurend op zoek naar betere, snellere en goedkopere ICT-oplossingen.

Wij willen modulair automatiseren en voeren 'ontkoppelen' hoog in het vaandel. Ontkoppelen heeft een aantal facetten. Eén van die facetten is het afkoppelen van bepaalde KPN-brede legacy-systemen die 'te groot en te duur' zijn voor onze BU. Andere systemen moeten worden aangekoppeld. Bij nieuwe systemen hanteren we als richtsnoer:
– kopen; – schaalbaarheid; – zo min mogelijk aanpassingen; – koppelbaarheid (minimale interface-complexiteit).
Een derde facet van 'ontkoppelen' is het voorkomen dat dezelfde functionaliteit in meerdere systemen voorkomt ('ontdubbelen'). Om een en ander te realiseren maken we sinds 2 jaar gebruik van de message broker ENGIN. Dat bevalt ons zeer goed. De interface-complexiteit is significant afgenomen. Nieuwe interfaces met reeds gekoppelde systemen kunnen veel sneller worden gerealiseerd. Zo bedroeg de doorlooptijd van het realiseren van een interface met ons systeem M&M007 voorheen ongeveer 7 maanden. Tegenwoordig is een en ander binnen 4 weken operationeel.

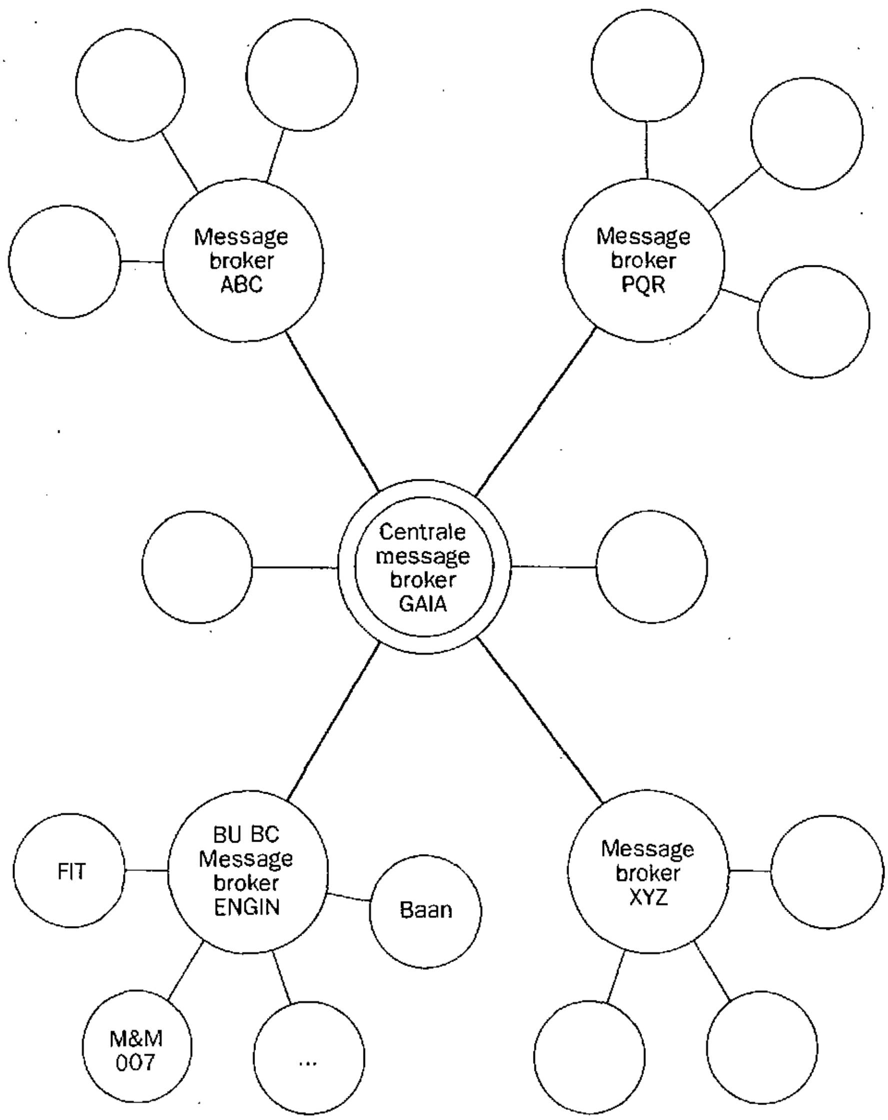
Door de nieuwe werkwijze kunnen wij niet alleen veel sneller reageren op verzoeken uit de gebruikersorganisatie, maar zijn ook de omvangrijke interfacekosten duidelijk afgenomen. Belangrijk is ook dat lijnmanagers die een nieuwe interface willen laten vervaardigen expliciet moeten aangeven welke kostenbesparingen met deze interface worden bereikt.
Wat de message broker betreft: wij zetten de gekozen koers met kracht voort!".

6 Conclusie

In hoofdstuk 2 is als probleemstelling geformuleerd:
"Er zijn teveel 1 op 1 interfaces en het onderhoud van de interfaces brengt te hoge kosten met zich mee". Het probleem kon als volgt worden aangepakt:

1 realiseer zoveel mogelijk nieuwe interfaces via de message broker;
2 vervang bij groot onderhoud van de huidige interfaces deze door interfaces via de message broker, tenzij er door het management goed te keuren redenen zijn om dat niet te doen;
3 rapporteer over de voor- en nadelen van het gebruik van de message broker.

De message broker ENGIN van BU BC als domein binnen GAIA



figuur 12

De BU BC is sinds 1998 bezig met het realiseren van nieuwe interfaces via de message broker (punt 1). Bij groot onderhoud van bestaande interfaces worden deze vervangen door interfaces via de message broker, waarbij de betreffende interface generiek wordt gemaakt (punt 2). Dit artikel draagt bij aan punt 3.

De conclusie van de BU BC is dat het gebruik van de message broker, voorzien van een Esperanto-faciliteit, het aantal 1 op 1 koppelingen substantieel vermindert, de kosten van realisatie en onderhoud sterk doet afnemen en een adequaat beheer van de interfaces mogelijk maakt.

De Business Unit BedrijfsCommunicatie zet met betrekking tot de gegevensuitwisseling/interfacing van applicaties met kracht de ingezette koers voort.

De conclusie wordt toegelicht aan de hand van een aantal voordelen, nadelen en richtlijnen.

6.1 Voordelen

- een message broker is een applicatie die volledig ingericht is voor interface management waardoor het eenvoudiger, en goedkoper, is om interfaces op elkaar af te stemmen;
- door het toepassen van een message broker kan elke applicatie in zijn eigen formaat met de buitenwereld communiceren. Hierdoor wordt het eenvoudiger om applicaties aan te passen of te vervangen;
- als iedere applicatie nog maar één interface met de buitenwereld hoeft te hebben, betekent dit dat de architectuur van de gehele "enterprise" gaat veranderen. Het maximum aantal interfaces neemt af van n*(n-1) naar 2*n;
- bundeling van kennis op één plaats;
- onafhankelijk (centraal) beheer van de interfaces wordt mogelijk (en is gewenst);
- het is mogelijk om een grote mate van standaardisatie met betrekking tot het transformeren in te voeren;
- koppelingen op datalaag-niveau, die de data-integriteit in gevaar brengt, worden voorkomen.

6.2 Nadelen

- als de message broker onderuit gaat, ligt alle communicatie tussen de applicaties stil (single point of failure);
- toename van de onderlinge afhankelijkheid van applicaties;
- de neiging bestaat om functionaliteit, die in applicaties thuishoort, te verplaatsen naar de message broker.

6.3 Richtlijnen

- om het interface management goed te stroomlijnen is het nodig een beheerproces in te richten dat applicatieonafhankelijk is. Het beheerproces moet een leidende rol spelen bij het definiëren en ondersteunen van koppelingen en moet voldoende bevoegdheden hebben om een leidende rol in het interface management in te vullen. Daarnaast moet er commitment van alle betrokken partijen zijn om koppelingen via de message broker te laten lopen;
- de applicatie die "eigenaar" van de gegevens is, bepaalt de recordlayout van het bericht;
- de message broker kan een vraag opsplitsen in meerdere vragen indien het antwoord uit verschillende systemen moet komen. Vervolgens kunnen de antwoorddelen weer samengevoegd worden tot één antwoord. Men moet bij het gebruik van deze functionaliteit wel oppassen dat men geen business functionaliteit in de message broker neerlegt;
- als twee applicaties gegevens uitwisselen, is het belangrijk te weten wat het doel van de communicatie is en welke rol de betrokken applicaties hebben. Dit bepaalt welke acties uitgevoerd worden en wat de inhoud van het antwoordbericht kan zijn. Zonder deze informatie kan een message broker zijn rol als "tussenstation" niet correct vervullen;
- de applicaties moeten bij voorkeur opgebouwd zijn volgens de drie-lagen-architectuur;
- de applicaties moeten er rekening mee houden dat sommige gegevens uit andere applicaties tijdelijk niet beschikbaar zijn;
- door adapters te bouwen kan, ten behoeve van communicatie, een functielaag worden gesimuleerd;
- koppelingen moeten uitgevoerd worden op het niveau van de functielaag. Datalaag-koppelingen zijn niet toegestaan;
- applicaties die sterk van elkaar afhankelijk zijn, moeten

tot hetzelfde domein behoren;
- uit oogpunt van continuïteit is het noodzakelijk om adequate maatregelen te nemen om de consequenties van een single point of failure te beperken;
- voor sommige bestaande maatwerkkoppelingen die veel business functionaliteit bevatten, kan het zinvol zijn deze niet via een message broker te koppelen.

Literatuurlijst

[Baan1999] Baan Development, *Baan EAI - Open World, Positioning White paper*, Baan Company, December 1999, 20 pag.

[Clau1999] Claus, M. *Inventarisatie, Functies en gegevens landelijke systemen BU BC*, KPN Telecom, BU BC, mei 1999, 37 pag. (intern rapport)

[Dec1998] Dec, K., R. Mack and E.Andren, *Transition Strategies, From Mainframes to Distributed Computing*, The Technical Issues, Gartner Group, February 1998, 76 pag.

[Dorr1999] Dorr, D. en J. Zuidema, *Werken met het model Nederlandse Kwaliteit, De Quick Scan van uw organisatie*, Kluwer Bedrijfsinformatie, 1999, 87 pag.

[Dijk1984] Dijk, A.J. van, R.P. Karlas, P. Makkes en R.J.J. van Schie, *'GEMS'- GEautomatiseerd Magazijnadministratie- en voorraadbeheerSysteem, Functioneel ontwerp*, Technische Universiteit Delft, Memorandum no RC-ISO-84 004, juni 1984, 256 pag.

[Dijk2000] Dijk, A.J. van, R.E. Overeem, Th. Stolker en J. Weber, *Interface Baan -ENGIN, Technisch Ontwerp*, KPN Telecom / Baan Company, april 2000, 24 pag. (intern rapport)

[Ecks1999] Eckstein, Robert, *XML*, Pocket Reference, O'Reilly, Cambridge, oktober 1999, 107 pag.

[GAIA2000] Projectgroep GAIA, *Basisontwerp GAIA*, KPN Telecom, januari 2000, 47 pag. (intern rapport)

[Musc1999] Muscato, *ENGIN v4.0, Message Broker & WebEnabler*, Muscato Corporation, Altamonte Springs, 35 pag.

[Niel1989] Nielen, G.C., *Omgaan met georganiseerde informatiesystemen*, Handboek Informatica, A1500, Samsom Bedrijfsinformatie, 1989, 24 pag.

[Niel2000] Nielen, G.C., *Innovatie in de wereld*, IT-Monitor, nr. 1, 2000, 1 pag.

[Otze2000] Otzen, E., *Message broker ENGIN, M&M007 service order interface*, KPN Telecom, februari 2000, 58 pag. (intern rapport)

[Over1998] Overeem, R., *Message broker, De inzet van een message broker*, KPN Telecom, juli 1998, 38 pag. (intern rapport)

[Pitt1999]Pitts, Natanya, *XML*, Sybex, Soest, juni 1999, 543 pag.

[Rock2000] Rock-Evans, R., *Kunnen IDL en EAI samen?*, Software Release Magazine, februari 2000, pag. 11-12.

[Stee1997] Steenbergen, M.E. van, *Mapping, Een handvat voor beslissingen*, KPN Telecom, BU BC, mei 1997, 17 pag. (intern rapport)

auteur

Ir. A.J. van Dijk RE RI is werkzaam als onafhankelijk IT-adviseur/IT-auditor bij Avédé-Info b.v. te Zoetermeer. Hij is bij de Business Unit BedrijfsCommunicatie van KPN Telecom betrokken bij architectuurvraagstukken en bij de invoering van Baan ERP.

Ing. R.E. Overeem is werkzaam als senior software engineer bij Logica b.v. te Groningen. Hij is bij het Software Huis van KPN Telecom betrokken bij de realisering van message broker toepassingen.

J.J Schijf is werkzaam als hoofd ontwikkeling bij de Business Proces Group van de Business Unit BedrijfsCommunicatie van KPN Telecom. Hij heeft in 1998 het initiatief genomen om de message broker ENGIN aan te schaffen.

Publication 5

EAI en ERP met Baan Open World

De realisatie van een adapter in de praktijk van de

Business Unit BedrijfsCommunicatie van KPN Telecom

EAI and ERP with Baan Open World
The realisation of adapters at KPN Telecom

2001

It is fascinating to see how people
cooperating in an ICT project and together,
sometimes uphold a situation for months
or even years on end that nobody really wants
and achieve an end result that nobody
is waiting for.

Nico Beenker [2004]

This article previously appeared in:

Software Release Magazine, June 2001, pages 30-35

Abstract
This article is about the realisation of adapters between BaanERP and the Engin message broker. First, a brief introduction about the case study at KPN Telecom is provided and a framework for EAI is outlined. Next, the realisation of the Baan-Engin-Adapter comes up for discussion.

thema

Dit artikel gaat over de realisatie van adapters tussen BaanERP en de message broker Engin. Eerst wordt een korte inleiding over de casus bij KPN Telecom gegeven en wordt een kader voor EAI geschetst. Daarna komt de realisatie van de Baan-Engin-Adapter aan de orde.

EAI en ERP met Baan Open World

De realisatie van een adapter in de praktijk van de Business Unit BedrijfsCommunicatie van KPN Telecom

KPN Telecom is in Nederland een zeer belangrijke speler op de telecommunicatiemarkt en moet snel en flexibel inspelen op de dynamiek in de telecommunicatiemarkt en de steeds veranderende wensen van de klanten. KPN doet dat door een gedifferentieerd producten- en dienstenaanbod. Daartoe is de organisatie van KPN ingedeeld in onder andere Business Units, Verkoopeenheden en Productie Eenheden. Eén van de Business Units is BedrijfsCommunicatie (BU BC). De BU BC houdt zich bezig met de inkoop, verkoop, installatie en service van alle apparatuur en systemen die bij klanten worden geplaatst. In het werkveld van de BU BC zijn enkele duizenden mensen werkzaam: veel monteurs, planners en werkvoorbereiders maar ook marketeers, financiële medewerkers en personeelsfunctionarissen. Het doel van de BU BC is marktleider zijn en blijven en het realiseren van een goed rendement. Om dat doel te bereiken werkt de BU continu aan het verbeteren van de bedrijfsprocessen en de daarbij ondersteunende ICT. De organisatiestructuur van KPN laat toe dat Business Units een zelfstandig IT-beleid voeren, mits er gebruik gemaakt wordt van corporate data, zoals klantgegevens.

Sinds augustus 2000 wordt door de business line BedrijfsCommunicatieSystemen (BCS) van de BU BC gewerkt aan het inrichten en implementeren van BaanERP software. Dat gebeurt binnen het project Mecano (Met ERP-Concept Aangestuurde Nieuwe Organisatie). BaanERP zal niet alle bestaande informatiesystemen vervangen. Daarom is het noodzakelijk dat BaanERP op adequate wijze communiceert met een aantal van de overige bedrijfsinformatiesystemen. Binnen Mecano is het Kolibrie-team (KPN On Line Interfacing met BaanERP. Resultaat Is Effectief) verantwoordelijk voor de realisatie van deze communicatie door middel van interfaces. In dit artikel staat de interface van BaanERP met het Centrale KlantRegistratiesysteem (CKR) en de wijze waarop deze met behulp van een message broker en een adapter is gerealiseerd, centraal.

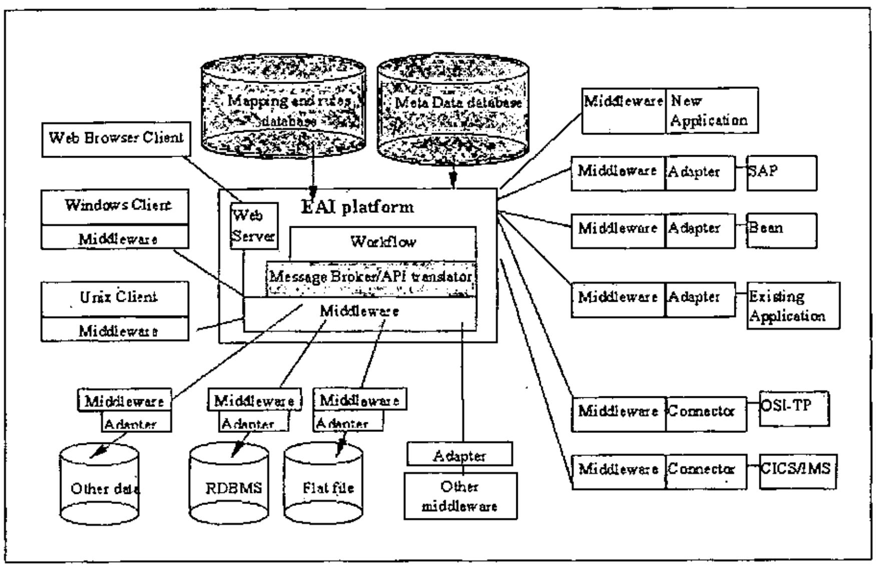
EAI Applicatie-integratie komt in verschillende vormen voor. Deze diversiteit maakt dat een standaard integratieoplossing niet bestaat. Enkele vormen van applicatie-integratie zijn [Elsw2001]:

- gegevens uit verschillende gegevensbronnen moeten worden gecombineerd om nieuwe gegevens af te kunnen leiden. Applicatie-integratie krijgt dan de vorm van gegevensintegratie. Dat is het gebied van de datawarehouses;
- applicaties moeten samenwerken om nieuw resultaat te kunnen bereiken. Daarbij wordt de functionaliteit van verschillende systemen geïntegreerd;
- applicaties vormen onderdeel van een keten. Ieder heeft haar eigen functionaliteit, maar de resultaten of gemeenschappelijk benodigde gegevens moeten worden doorgegeven aan de volgende applicatie in het bedrijfsproces. In deze context wordt vaak de term enterprise application integration (EAI) gebruikt. De keten kan ook over de grenzen van een organisatie heen gaan, bijvoorbeeld richting andere bedrijven (Business to Business).

In dit artikel komt applicatie-integratie in de laatste vorm (EAI) aan de orde. Applicatie-integratie vereist dat de uiteenlopende representaties uit verschillende bronnen op elkaar kunnen worden afgebeeld. Dit vereist een soort gemeenschappelijke taal waarin vorm en betekenis van de gemeenschappelijke concepten worden vastgelegd: semantische integratie. Een actuele syntactische (vorm) taal is XML (eXtensible Markup Language).

Vanuit de technologische invalshoek is EAI ook het proces waarmee, via een netwerk dat de machines van een of meer bedrijven met elkaar verbindt, zowel applicaties als gegevensbestanden kunnen worden geïntegreerd. De applicaties en de gegevensbestanden kunnen gebouwd zijn met een diversiteit aan technologieën. Daarbij vereist de integratie weinig tot geen wijzigingen van de bestaande applicaties en/of gegevens, oftewel een 'inbreukvrije' benadering [Rock2000]. EAI ontleent zijn enorme belang aan het feit dat het een vorm van op componenten gebaseerde ontwikkeling mogelijk belooft te maken. De belangrijkste componenten van een EAI-architectuur zijn (zie figuur 1):

- middleware: de middleware levert de onderliggende ondersteuning voor gedistribueerde verwerking en zorgt door het gehele netwerk heen voor de veilige en betrouwbare aflevering van berichten en aanroepen;
- adapters en connectoren: adapters zorgen voor de wederzijdse vertaling tussen de verschillende technologische niveaus, zoals het vertalen van de aanroepen berichtopmaak van de intern gerichte opmaak naar de opmaak zoals die door een door derden geproduceerd product wordt ondersteund en vice versa. Connectoren worden gebruikt om transactieverwerkingsprotocollen te vertalen;
- message brokers: Gartner [Dec1998] geeft als definitie en doel van de message broker:
  "Message brokers are logical hubs that copy and forward messages to one or more destinations. A message broker is an intelligent third party (hence a 'broker') between information sources and information consumers. The purpose of this architectural type is to integrate independently designed application domains (e.g., new applications, purchased applications and legacy applications). Message brokers make it possible to re-engineer business processes without re-engineering all the application programs and databases";
- workflow: de workflowsoftware bestuurt het proces. Dit houdt bijvoorbeeld in dat een gebruiker bepaalde informatie invoert, waarna de workflow zorgt voor het activeren van de message broker, applicaties en adapters.



FIGUUR 1: Componenten van een EAI-architectuur (bron: Rock-Evans)

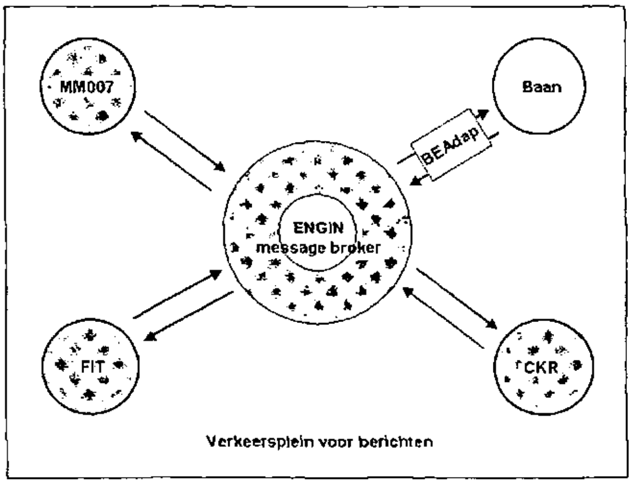
INTERFACES Een mogelijke definitie van een interface is: een koppeling tussen (twee) applicaties met als doel het uitwisselen van gegevens. Het ontwikkelen van een interface kan worden gezien als het ontwikkelen van een beperkt informatiesysteem. Daarbij worden de volgende stappen onderkend:

- indienen van een verzoek om een interface te realiseren;
- houden van een intake-gesprek;
- uitvoeren een impactanalyse;
- opstellen van een Functioneel Interface Document;
- opstellen van een Technisch Ontwerp;
- realiseren;
- testen;
- installeren;
- beheren.

Een complicerende factor bij interfaces is dat er altijd meerdere partijen bij betrokken zijn.

INVULLING EAI BIJ HET PROJECT MECANO Binnen de BU BC en het project Mecano wordt EAI onder andere ingevuld met de message broker Engin van de firma Muscato/Optio Software. Om BaanERP te koppelen met de overige bedrijfsinformatiesystemen wordt gebruik gemaakt van Baan Open World en van adapters.

De BU BC heeft de message broker Engin drie jaar geleden aangeschaft. Belangrijke motieven daarvoor waren enerzijds het beter beheersbaar te maken van het beheer van interfaces en anderzijds het sterk omlaag brengen van de kosten. Met de komst van de message broker is een proces ingezet, waarbij nieuwe interfaces zoveel mogelijk via de message broker worden gerealiseerd en bestaande interfaces bij groot onderhoud worden vervangen door interfaces via de message broker. Op die manier moet het aantal 1 op 1 interfaces, die hoge kosten met zich meebrengen, sterk worden teruggebracht. Over de ervaringen met Engin is zeer recent een uitvoerig artikel gepubliceerd [Dijk2001]. Naast de con-



FIGUUR 2: Message broker Engin en de Baan-Engin-Adapter binnen de EAI van de BU BC

clusies over het gebruik zijn hierin voordelen, nadelen en richtlijnen gegeven.

Enkele richtlijnen zijn:

- het is niet gewenst om business functionaliteit in de message broker te beleggen;
- applicaties moeten bij voorkeur zijn opgebouwd volgens de drie-lagen-architectuur;
- koppelingen van applicaties moeten worden uitgevoerd op het niveau van de functielaag;
- door adapters te bouwen kan, ten behoeve van communicatie, een functielaag worden gesimuleerd.

Figuur 2 geeft een (beperkte) weergave van de EAI-architectuur van de BU BC. De message broker Engin fungeert als "verkeersplein voor berichten". BaanERP communiceert via een Baan-Engin-Adapter (kortweg BEAdap) en Engin met andere applicaties zoals CKR.

Een derde component binnen de EAI-architectuur is Baan Open World (kortweg BOW). BOW is een integratie-framework om Baan Enterprise Solutions applications, waaronder BaanERP, onderling te laten communiceren of te laten communiceren met andere applicaties. Binnen de EAI van BU BC wordt BOW gebruikt om BaanERP te laten communiceren, via Engin en BEAdap's, met de overige bedrijfs-informatiesystemen. BOW bestaat onder andere uit de volgende componenten (zie figuren 3 en 5):

- een aantal BOW-adapters, waaronder een BaanERP Open World Adapter;
- Business Object Interfaces (BOI's). Een BOI is een standaardmethode (API) om Baan Open World objecten via de BOW-adapters te verbinden met andere applicaties. Een BOI maakt het mogelijk om op een andere manier dan via de online BaanERP-sessies, met behoud van de BaanERP-functionaliteit, te communiceren met BaanERP. Binnen BaanERP bestaat een Business Object uit een of meer gegevenstabellen en programma's (DAL, DLL) om de gegevens te bewerken. BaanERP heeft een aantal standaard BOI's. Indien nodig kunnen aanvullende BOI's worden gemaakt;
- Baan Open World Studio, de ontwikkelomgeving om BOI's te creëren;
- XML, dat als hulpmiddel wordt gebruikt om koppelingen tussen applicaties te vereenvoudigen of stroomlijnen;
- Baan Open World Services, waaronder de transportservice (op O/S-laag en netwerklaag) waarmee het berichtenverkeer tussen twee applicaties wordt geregeld.

Baan Open World is aanvullend op middleware en is ontworpen om te communiceren met een groot aantal transportprotocollen en producten.

De Baan-Engin-Adapter voor CKR

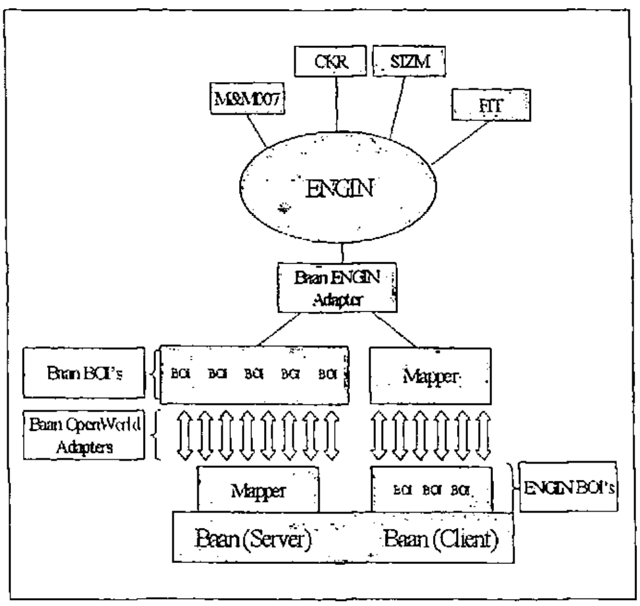
Functioneel Ontwerp van de interface BaanERP - CKR

CKR is een corporate informatiesysteem waarmee binnen KPN Telecom klantgegevens worden geregistreerd. Andere informatiesystemen kunnen gebruik maken van deze centraal beheerde klantgegevens. Daartoe heeft CKR een aantal transacties beschikbaar zoals:

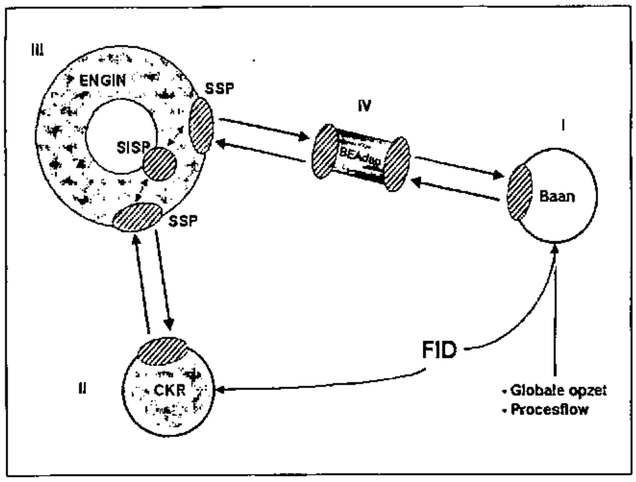
- klantgegevens opvragen uit CKR;
- het aanbrengen van een verwijzing bij een klant. Daarmee wordt een abonnement op mutaties genomen. Dagelijks worden gegevensmutaties van de klanten waarop een abonnement is genomen via batchverwerking door CKR aan de abonnees aangeboden;
- het verwijderen van een verwijzing.

Indien gewenste klantgegevens nog niet in BaanERP zijn opgenomen, dan worden deze gegevens met behulp van een interface tussen BaanERP en CKR opgevraagd bij CKR en opgeslagen in BaanERP.

Hoe komt de interface tot stand? Na het uitvoeren van een impactanalyse wordt een Functioneel Interface Document (FID) opgesteld. Naast een korte inleiding van het betreffende proces komen gegevensoverdracht en selectiecriteria aan de orde. Het FID is het Functioneel Ontwerp van de interface. Het FID wijkt af van het Functio-



FIGUUR 3: Message broker Engin en Baan Open World



FIGUUR 4: De interface BaanERP - CKR

neel Ontwerp van een "gewoon" informatiesysteem doordat het bij een interface gaat om meer dan één informatiesysteem. In dit geval gaat het om een interface (gegevensuitwisseling) tussen CKR en BaanERP. In figuur 4 is de positie van het FID aangegeven. Het FID is bij deze interface uitgebreid met de documenten "Globale opzet" en "Procesflow". Deze documenten beschrijven hoe met de klantgegevens in BaanERP wordt omgegaan.

TECHNISCH ONTWERP VAN DE INTERFACE BAANERP - CKR Het Technisch Ontwerp van een interface bestaat uit een aantal componenten (zie figuren 4 en 5):

- een technisch deelontwerp van de vragende applicatie, in dit geval BaanERP,
- een technisch deelontwerp van de leverende applicatie, in dit geval CKR,
- een technisch ontwerp van de oplossing binnen de message broker Engin,
- een technisch ontwerp van de Baan-Engin-Adapter/CKR.

Figuur 4 laat zien dat alle gearceerde delen een bijdrage leveren aan het Technisch Ontwerp.

BAANERP Zoals aangegeven is het FID van de interface BaanERP-CKR uitgebreid met een "Globale opzet". Hierin is met name beschreven hoe binnen BaanERP moet worden omgegaan met CKR/klantgegevens. Door de BU BC worden binnen BaanERP aanvullende klantgegevens, zoals een CKR-nummer opgeslagen. Sommige van deze gegevens worden als zoekargument gebruikt. Dit heeft geleid tot aanpassingen in BaanERP. Daartoe is een technisch deelontwerp gemaakt.

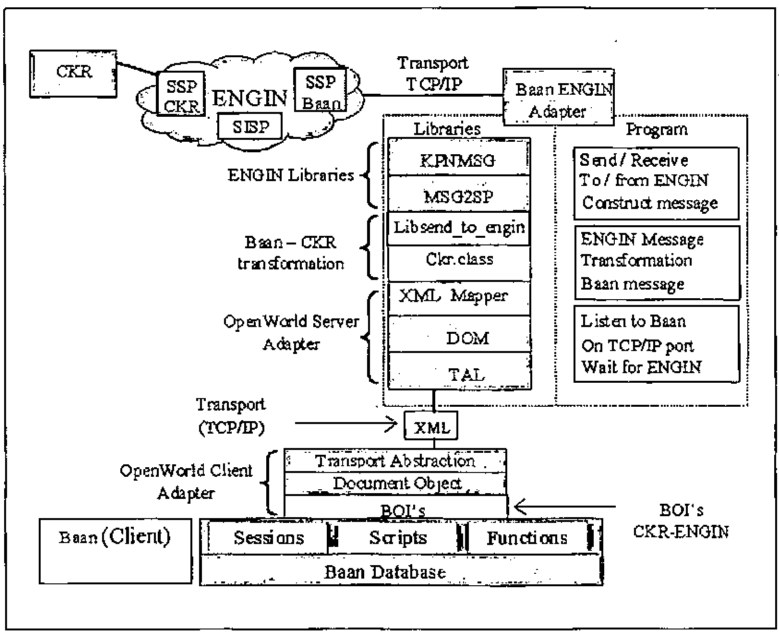
CKR CKR is gemaakt om diensten betreffende klantgegevens te verlenen. Daartoe is een aantal standaardtransacties gedefinieerd waarmee de diensten kunnen worden afgenomen. In feite kunnen deze transacties worden beschouwd als CKR-BOI's. De interface BaanERP-CKR is gerealiseerd met behulp van een viertal CKR-BOI's. Aanpassingen aan CKR, bijvoorbeeld een extra transactie/BOI, waren niet nodig.

ENGIN De communicatie van applicaties met de message broker Engin verloopt via een Service SPecificatie (SSP). In een SSP wordt vastgelegd welke gegevens worden aangeboden en welke controles en transformaties moeten plaatsvinden. In feite kunnen SSP's worden gezien als Engin-BOI's.

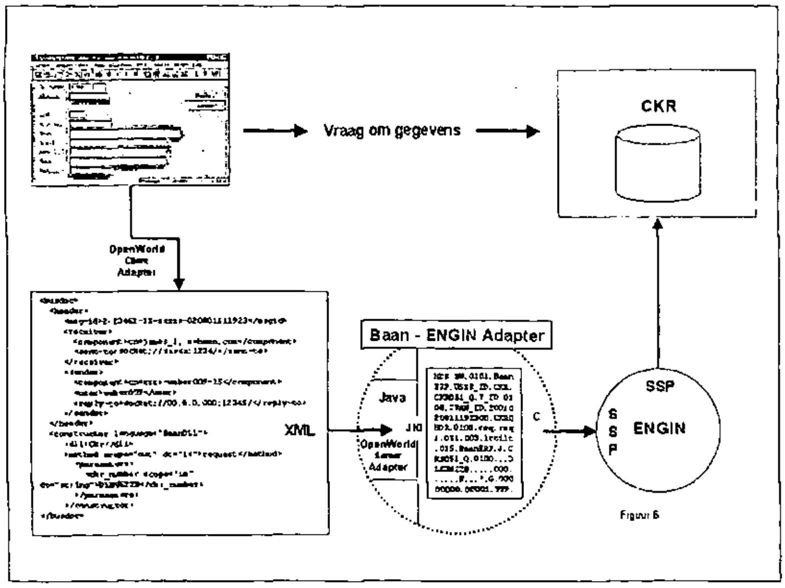
Engin zorgt voor het afbeelden van de gegevens van de ene applicatie op die van de andere applicatie. Daartoe wordt een Service Implementatie SPecificatie (SISP) opgesteld.

BEADAP/CKR De BEAdap/CKR zorgt voor een wederzijdse vertaling tussen Engin en BaanERP. Binnen de versie van Engin die bij de BU BC in gebruik is, wordt nog niet gewerkt met XML, maar met segmenten als beschrijvingsmethode. BOW is gebaseerd op XML. In de volgende paragraaf wordt in vogelvlucht de werking van de BEAdap/CKR beschreven.

DE WERKING VAN DE BEADAP/CKR Het realiseren van de BEAdap/CKR was een interessante activiteit omdat de twee omgevingen Engin en BaanERP nog niet eerder met elkaar hadden samengewerkt. Engin fungeert bij de BU BC al enkele jaren op basis van segmenten. ERP systemen zijn lange tijd gesloten geweest. Alle belangrijke bedrijfsgegevens werden in ERP gestopt. De laatste tijd bestaat bij veel bedrijven de behoefte om ERP-systemen te combineren met Best of Breed applicaties of eigen applicaties. BOW is ontwikkeld om te kunnen voldoen aan de door klanten noodzakelijk geachte communicatie van BaanERP met andere informatiesystemen. KPN Telecom



FIGUUR 5: Technisch ontwerp BaanERP - CKR



FIGUUR 6: Flow verzoek van BaanERP aan CKR

is één van de eerste bedrijven die gebruik maakt van BOW. Na uitvoerige discussies is besloten om de adapter niet onder te brengen bij Engin of BaanERP, maar als een afzonderlijk proces te realiseren. Belangrijke kwaliteitsaspecten daarbij waren flexibiliteit en aanpasbaarheid. Verwacht mag worden dat nieuwe ontwikkelingen, zoals XML, nieuwe (vereenvoudigde) communicatiemogelijkheden bieden. Het vervangen van een afzonderlijk proces is dan eenvoudiger dan een proces dat sterk verweven is met de message broker of met BaanERP. De BEAdap bestaat uit de volgende componenten (zie figuur 5):

- een mechanisme om naar Engin te luisteren (het "Engin-oor");
- een mechanisme om naar BaanERP te luisteren (het "BaanERP-oor");
- een deel (deel 1) dat tot doel heeft berichten van BaanERP te ontvangen of aan BaanERP aan te bieden;
- een deel (deel 3) dat tot doel heeft berichten van Engin te ontvangen of aan Engin aan te bieden;
- een deel (deel 2) dat de communicatie (gegevens transformeren en doorgeven) tussen deel 1 en deel 3 verzorgt.

Op een aantal punten wordt nu kort ingegaan.

XML De bij de BU BC werkende versie van Engin werkt nog niet met XML. Om de gegevensstructuur van de Engin-BOI (de SSP) bekend te maken binnen de BEAdap en binnen BaanERP wordt een opgeleverde SSP omgeschreven naar een XML-bericht. Dit bericht wordt door een XML-mapper vertaald naar een Java-omgeving ten behoeve van deel 1 van de BEAdap en naar een Baan 3 GL-omgeving ten behoeve van BaanERP. Op die manier is de gegevensstructuur bekend binnen alle componenten.

DEEL 3 - ENGIN Het Engin-deel bevat de Engin Libraries. De functionaliteit in deze libraries is ontwikkeld door het Software Huis van KPN en is geschreven in de programmeertaal C.

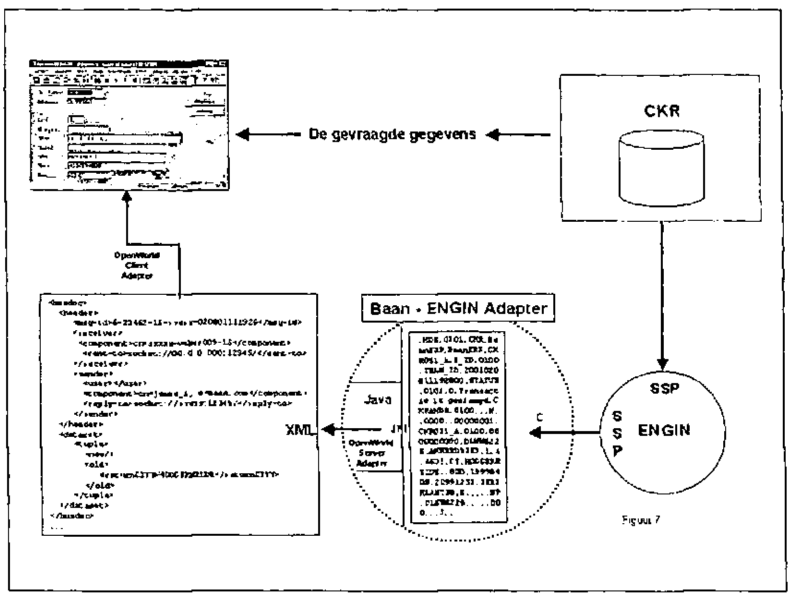
DEEL 1 - BAANERP Het BaanERP-deel bevat (een deel van) de BOW-adapters. De gehanteerde programmeertaal is Java. De BOW-adapters communiceren op basis van XML met BOW-adapters aan de BaanERP kant Afhankelijk van de initiatiefnemer vervullen ze de rol van BOW-server-adapter of BOW-client-adapter.

DEEL 2 De communicatie tussen deel 1 (Java) en deel 3 (C) gebeurt met behulp van Java Native Interface (JNI).

ENGIN-OOR Het Engin-oor is een deelproces dat luistert op een poort naar berichten die verwerkt moeten worden. Deze functionaliteit wordt beschikbaar gesteld door de Engin Libraries.

BAANERP-OOR Het BaanERP-oor is een deelproces dat luistert op een poort naar XML-berichten vanuit BOW. Deze functionaliteit maakt deel uit van de BOW-adapters.

DE FLOW VAN HET OPVRAGEN EN VERWERKEN VAN EEN CKR-BERICHT Figuur 6 laat zien hoe een verzoek vanuit een BaanERP-sessie aan CKR via de BEAdap/CKR en Engin verloopt. Het CKR-nummer van de klant wordt door de betreffende sessie van BaanERP aangeboden aan BOW. De BOW-client-adapter biedt een XML-



FIGUUR 7: Flow gegevens van CKR naar BaanERP

Afkortingen/acroniemen

API	Application Programming Interface
BEAdap	Baan-Engin-Adapter
BO	Business Object
BOI	Business Object Interface
BOW	Baan Open World
BU BC	Business Unit BedrijfsCommunicatie
CKR	Centrale KlantenRegistratie
DAL	Data Access Layer
DCD	Document Contents Description
DLL	Dynamic Link Library
DOM	Document Object Module
EAI	Enterprise Application Integration
ERP	Enterprise Resource Planning
FID	Functioneel Interface Document
ICT	Informatie- en CommunicatieTechnologie
JNI	Java Native Interface
Kolibrie	Kpn On Line Interfacing met BaanERP. Resultaat Is Effectief
Mecano	Met Erp-Concept Aangestuurde Nieuwe Organisatie
QCD	Query Contents Definition
SISP	Service Implementatie SPecificatie
SSP	Service SPecificatie
XML	eXtensible Markup Language
TAL	Transport Abstraction Layer

bericht aan op een poort. De BOW-server-adapter, die onderdeel uitmaakt van de BEAdap/CKR, luistert naar deze poort en pikt het bericht op. Met behulp van een Java-programma wordt, via JNI, het CKR-nummer doorgegeven aan een C-programma dat de gegevens aanbiedt aan een (Engin-)poort. De message broker pikt de gegevens op en stuurt deze naar de applicatie CKR.

Figuur 7 toont hoe de gegevens vanuit CKR via Engin en de BEAdap/CKR op het scherm van de BaanERP-gebruiker komen. CKR zet de gegevens die behoren bij het betreffende CKR-nummer klaar voor Engin. De message broker pikt de gegevens op en geeft ze, via de (Engin-)poort door aan de BEAdap/CKR. Het C-programma stuurt de gegevens, via JNI, door naar de BOW-server-adapter. Deze biedt de gegevens door middel van XML-berichten aan op de poort. De BOW-client-adapter pikt de XML-berichten op en via een sessie van BaanERP komen de CKR-gegevens op het scherm van de BaanERP-gebruiker. De gebruiker kan ze, na een visuele controle, met behulp van de BaanERP-sessie in de database opslaan.

Conclusie Het neerzetten van een infrastructuur waarmee de communicatie tussen BaanERP en CKR adequaat verloopt, via Baan Open World en een BEAdap en de message broker Engin, heeft een substantiële inspanning vereist. Met name het op elkaar laten aansluiten van de diverse componenten heeft de nodige inspanning gevraagd en zal verder worden geoptimaliseerd. Nieuwe BEAdap's kunnen tamelijk snel worden gerealiseerd omdat de BEAdap/CKR hergebruikt kan worden. Alleen het datadeel moet worden aangepast. Figuur 5 geeft aan welke onderdelen dat zijn (de witte onderdelen). Inmiddels zijn enkele BEAdap's gerealiseerd. Het komende jaar zal nog een aantal BEAdap's in gebruik worden genomen. De verwachting bestaat dat nieuwe versies van Engin, gebaseerd op XML, en BOW de communicatie zal vereenvoudigen.

Ir. Aart J. van Dijk RE
is werkzaam als onafhankelijk IT-adviseur/IT-auditor bij Avédé-Info b.v. te Zoetermeer. Hij is bij de Business Unit BedrijfsCommunicatie van KPN Telecom betrokken bij architectuurvraagstukken en bij de invoering van BaanERP.

Drs. Jeroen Weber
was tot voor kort werkzaam als Senior International Technical Consultant bij Baan Nederland N.V. en was bij KPN Telecom betrokken bij de invoering van BaanERP. Hij is thans werkzaam bij SeeBeyond te Hoofddorp.

Literatuur

[Baan1999] Baan Development,
Baan EAI – Open World, Positioning White paper, Baan Company, December 1999

[Dec1998] Dec, K., R. Mack and E.Andren, Transition Strategies, From Mainframes to Distributed Computing, The Technical Issues, Gartner Group, February 1998

[Dijk2001] Dijk, A.J. van, R.E. Overeem en J.J. Schijf, Gegevensuitwisseling (interfacing) van applicaties met behulp van een message broker, De applicaties van de Business Unit BedrijfsCommunicatie van KPN Telecom, IT Management [Select], jaargang 7, nummer 1, mei 2001, pp. 39-57

[Elsw2000] Elswijk, M. van en M. Maat, Aspecten van een integratiestrategie, Informatie, november 2000, pp. 44-48

[Musc1999] Muscato/Optio Software, ENGIN v4.0, Message Broker & WebEnabler, Muscato Corporation, Altamonte Springs, 1999

[Rock2000] Rock-Evans, R., Kunnen IDL en EAI samen?, Sofware Release Magazine, februari 2000, pag. 11-13.

[Wage2001] Wagemans, R. en M. Baan, Functioneel Interface Document BaanERP —> CKR, Project Mecano, deelproject Kolibrie, KPN Telecom, februari 2001 (intern rapport)

[Webe2001] Weber, J. and A.J. van Dijk, Technical Design Interface Baan - CKR, Project Mecano, subproject Kolibrie, KPN Telecom, March 2001 (intern rapport)

Publication 6

Doorbelasting van kosten van gebruik en beheer van corporate data

De basisregistraties bij het Gak/ASZ

Charging costs of use and management of corporate data
The national databases of employees,
employers and employments

1998

An integrated approach helps us achieve
an overall understanding.

Tarek Abdel-Hamid & Stuart Madnick [1991]

This article previously appeared in:

IT Management [Select], December 1998, pages 61-78

Abstract
Charging of costs is a subject that gets regular attention within Information Technology. However, an adequate solution is not always immediately available. This article deals with the charging of costs that are associated with making available, maintaining and using of corporate data with the GAK Group. ASZ (Automatisering Sociale Zekerheid), the regular supplier of computerisation services of the GAK Group, manages the corporate databases and corresponding basic register systems. Within the GAK Group, there was a crying need for arriving at a different method of charging. People felt in particular the need for no longer basing the charging on the traditional performance units but instead on for users comprehensible units. This article describes how the ASZ Productteam Basisregistraties (PB) fleshed out this need. The BasisRegistratie Personen is used as an example.

Aart J. van Dijk

DOORBELASTING VAN KOSTEN VAN GEBRUIK EN BEHEER VAN CORPORATE DATA

samenvatting

Dit artikel handelt over de doorbelasting van de kosten die gepaard gaan met het beschikbaarstellen, beheren en gebruiken van bedrijfsgegevens binnen de GAK-Groep. ASZ Automatisering sociale zekerheid (ASZ), de huisleverancier van automatiseringsdiensten van de GAK-Groep, beheert de corporate databases en de bijbehorende basisregistraties. Binnen de GAK-Groep was er grote behoefte om tot een andere wijze van doorbelasten te komen. Met name werd de behoefte gevoeld om de doorbelasting niet langer op de klassieke prestatie-eenheden te baseren, maar op voor de gebruikers begrijpelijke eenheden. Dit artikel beschrijft hoe het Productteam Basisregistraties van ASZ invulling aan deze behoefte heeft gegeven. De BasisRegistratie Personen wordt als voorbeeld gebruikt. Dit sluit goed aan op het in 1995 in IT Management [Select] gepubliceerde artikel 'Gedistribueerde databases in de praktijk: de BasisRegistratie Personen bij het GAK'.

1 Inleiding

Doorbelasting van kosten is een onderwerp dat binnen de IT regelmatig in de belangstelling staat. Een adequate oplossing is echter doorgaans niet direct voorhanden. Bovendien bestaat er niet één allesomvattende oplossing. Dit artikel handelt over de resultaten van een onderzoek naar de verbetering van de doorbelasting van de kosten die gepaard gaan met het beschikbaarstellen, het beheren en het gebruiken van de corporate gegevens die behoren tot de basisregistraties. De gegevens zijn eigendom van GAK-Nederland (het GAK), en worden in technische zin beheerd (bewaard) door het Productteam Basis Registraties (PB) van ASZ Automatisering sociale zekerheid (ASZ).

IT Management [Select] heeft, in het kader van gedistribueerde databases, in 1995 reeds aandacht besteed aan de BasisRegistratie Personen (BRP).

In paragraaf 2 wordt ingegaan op de probleemstelling en het doel van het verrichte onderzoek. Het doorbelasten van de basisregistraties op basis van prestatie-eenheden of licenties levert een aantal knelpunten op. Het doel van het onderzoek is om tot een wijze van doorbelasten van de kosten van het beheer en gebruik van de basisregistraties te komen, die beter aansluit bij de bedrijfsprocessen van de afnemers.

Eerst wordt de positie van het PB binnen het GAK beschreven. Deze positie is van belang bij de in dit artikel genoemde doorbelastingsketen.

GAK-Groep

De GAK-Groep is een belangrijke speler op het complexe en veelzijdige terrein van de sociale zekerheid. De Groep bestaat uit een publieke en een private tak. De publieke tak, het GAK, voert de administratie uit voor het Landelijk Instituut Sociale Verzekeringen (LISV), dat de taken heeft overgenomen van de bedrijfsverenigingen. Deze bedrijfsverenigingen, voorheen de opdrachtgevers van het GAK, zijn in het kader van de nieuwe Organisatiewet Sociale Verzekeringen (OSV) per 1 maart 1997 opgeheven. Branches kunnen zich voortaan laten vertegenwoordigen door zogeheten sectorraden. De werkzaamheden van het GAK omvatten hoofdzakelijk de premie-inning en uitkeringsverzorging voor de Algemene Arbeidsongeschiktheidswet(AAW), de Wet op de Arbeidsongeschiktheidsverzekering (WAO), de Werkloosheidswet (WW), de Toeslagenwet (TW) en het niet geprivatiseerde deel van de Ziektewet (ZW). In de private tak van de GAK-Groep, onder de GAK-Holding, zijn de volgende commerciële werkmaatschappijen samengebracht:

- GAK-Diensten (verzekeringen en administratie);
- ASZ;
- Svianed (netwerkdiensten);
- Arbo Groep GAK (arbodienstverlening);
- ViaPrisma (administratieve lastenverlichting);
- BSN (bedrijfshulpverlening & security management).

ASZ

ASZ is als automatiseringsdochter binnen de GAK-Groep op 22 december 1995 te Amsterdam opgericht en is gespecialiseerd in de combinatie sociale zekerheid en automatisering. ASZ ontwikkelt, onderhoudt en exploiteert informatiesystemen. De GAK-Groep is de belangrijkste klant. Hoogwaardige informatietechnologie is van levensbelang voor een efficiënte beheersing van de zeer complexe informatiestromen en administratieve processen die inherent zijn aan de uitvoering van sociale wetten. Vanaf 14 duizend werkplekken wordt dagelijks met de systemen van ASZ gewerkt. De zeer omvangrijke gegevensbestanden behoren tot de grootste databases van Nederland.

ASZ biedt zijn klanten een samenhangend pakket diensten en producten. Kenmerk van de portfolio is dat het aanbod als totaalpakket kan worden aangeboden, maar dat bovendien steeds, afhankelijk van de situatie van de klant, met behulp van bestaande diensten en producten maatwerk kan worden ontwikkeld. De diensten zijn onder te verdelen in de volgende groepen:

- projectmanagement en consultancy;
- ontwikkeling, vernieuwing en implementatie van informatiesystemen;
- functioneel onderhoud van applicaties;
- processing en exploitatie;
- opleiding en training.

ASZ is een bedrijf met meer dan 700 werknemers en bestaat uit een aantal stafdiensten en vier business units. Binnen de business unit *systeemhuis* opereren zes afdelingen, waaronder het Productteam Basis Registraties (PB).

2 Probleemstelling

De basisregistraties vormen het belangrijkste aandachtsgebied van het PB. Binnen de informatie-architectuur van het GAK vormen de volgende drie basisregistraties een belangrijk deel van de gegevensinfrastructuur (corporate data):

- de basisregistratie *personen* (BRP) bevat de persoonsgegevens van alle verzekerden waarmee het GAK relaties onderhoudt;
- de basisregistratie *werkgevers* (BRWG) bevat de gegevens van werkgevers waarmee het GAK ten behoeve van de administratie van haar opdrachtgevers relaties onderhoudt of onderhouden heeft;
- de basisregistratie *dienstverbanden* (BRDV) bevat de huidige en historische dienstverbanden tussen personen en werkgevers. De dienstverbandgegevens blijven tot vijf jaar na beëindiging van het dienstverband opvraagbaar.

De gegevens die deel uitmaken van de drie basisregistraties (kortweg: basisgegevens) zijn opgeslagen in drie databases. Met betrekking tot de basisregistraties kunnen de volgende rollen worden onderscheiden:

- het GAK is eigenaar van de gegevens;
- het eigendomsrecht is overgedragen/gemandateerd aan de houder van de drie basisregistraties. De houder is onder andere verantwoordelijk voor de inhoudelijke kwaliteit van de gegevens;
- de gebruikers van de gegevens zijn alle geautoriseerde werknemers van het GAK;

- de exploitatie wordt verzorgd door het PB, dat daarbij ondersteund wordt door de afdeling Beheer Infrastructuur (BI) van ASZ;
- PB verzorgt het onderhoud van de informatiesystemen die behoren tot de basisregistraties.

PB fungeert als intermediair tussen de exploitatie enerzijds en de gebruikers en houders anderzijds. De kosten die gepaard gaan met de exploitatie en het gebruik van de basisgegevens (de zogenoemde werklast) worden door BI doorbelast aan PB, die de kosten vervolgens tezamen met de andere kosten (met name onderhoudskosten) doorbelast aan de houders en gebruikers. PB is voor de gebruikers van de basisregistraties het directe aanspreekpunt binnen ASZ.

In het verleden is doorbelast op basis van prestatie-eenheden die zijn afgeleid van de werklast. In 1997 is doorbelast op basis van zogenoemde BR-licenties. Op beide doorbelastingswijzen wordt nu kort ingegaan.

Prestatie-eenheden

Het begrip werklast kan simpel omschreven worden als 'dat wat een systeem te verwerken krijgt als gevolg van het verwerken van applicaties'. Werklast kan worden uitgedrukt in prestatie-eenheden: CVE-tijd, aantal in- en uitvoeropdrachten en geheugengebruik. In ASZ-termen: vups (vax unit of performance), mips en gigabytes.

Knelpunt prestatie-eenheden

Met betrekking tot de beschrijving van de werklast liggen de werelden van de afnemers van IT-diensten en de technici van rekencentra vaak ver uit elkaar. Terwijl IT-technici praten over vups, mips en gigabytes, willen de afnemers praten over hun bedrijfsprocessen.

BR-licenties

De gegevens van de basisregistraties (BR-gegevens) kunnen met de diverse informatiesystemen worden benaderd. Daarvoor is het nodig dat op het moment van benaderen toestemming is of kan worden verleend. Zo'n toestemming is een BR-aansluitinglicentie. Indien gebruikers op een andere manier dan via de primaire processystemen toegang tot de BR-gegevens willen hebben, is eveneens toestemming nodig. Een dergelijke toestemming heet BR-gebruikerslicentie. Er is in verband met het verschil in kosten, onderscheid gemaakt tussen een BR-aansluitinglicentie en een BR-gebruikerslicentie. Het aantal gebruikers dat tegelijkertijd actief kan zijn, is bepalend voor de omvang van de doorbelasting.

Knelpunten BR-licenties

Het werken met BR-licenties heeft geleid tot een aantal knelpunten:
- het leidt ertoe dat sommige gebruikers op een oneigenlijke, niet bedoelde manier gaan sturen. Ze maken namelijk afwegingen tussen het aanschaffen van een extra licentie of het aanpassen van de primaire processen. Op die manier werpen de licenties als het ware een drempel op. Dit strookt niet met de doelstelling van een optimale beschikbaarstelling van de basisregistraties;
- de doorbelasting via BR-licenties omvat slechts 77 procent van de doorbelastingskosten. De overige 23 procent wordt via applicaties op basis van verbruikte vups, mips en gigabytes doorbelast;
- mede door het werken met de client/server-technologie zal het steeds moeilijker worden om de prijs van BR-licenties vast te stellen;
- de prijs van de licenties is voor een groot deel gebaseerd op de prijs van de vup. Momenteel worden door BI veel indirecte kosten verdisconteerd in de vup-prijs (een 'dikke vup'). BI werkt aan het uitkleden van de vup ('dunne vup'), waardoor slechts een zeer beperkt deel van de basisvoorzieningen in de vup-prijs wordt opgenomen. Voorbeeld: met betrekking tot beveiliging zal in de vup-prijs een component worden opgenomen voor een standaardbeveiligingsniveau. Extra beveiligingsvoorzieningen zullen via afspraken in dienstenovereenkomsten worden geregeld en doorbelast.

Aanvullende eisen

Het GAK vindt dat ASZ zichtbaar aan kostenbeheersing en -verlaging moet werken. Tevens is het GAK van mening dat ASZ de doorbelastingsinformatie zodanig moet opleveren, dat zij de informatie kan gebruiken om de kosten toe te rekenen aan haar producten.

2.1 Doel en begrenzing van het onderzoek

Doel van het onderzoek was om tot een andere wijze van doorbelasten te komen van de kosten van het beheer (exploitatie en onderhoud) en het gebruik van de gegevensverzamelingen en applicaties die behoren tot de drie basisregistraties. Dit moet leiden tot:
- betere aansluiting bij de bedrijfsprocessen van de afnemers;
- grotere herkenbaarheid;
- grotere mogelijkheid om de omvang van de in rekening gebrachte kosten te beïnvloeden;

- beheersing van en (bij een gelijk verbruik) verlaging van de in rekening gebrachte kosten.

3 Huidige situatie

3.1 Doel van het proces beschikbaarstellen

PB stelt operationele applicaties betreffende de drie basisregistraties beschikbaar aan:
- de afnemers binnen de gebruikersorganisatie (gebruikers en houders/opdrachtgevers, zowel centraal als decentraal);
- de productteams die de primaire processystemen onderhouden;
- externe partijen buiten het GAK.

3.2 Beschrijving van het proces beschikbaarstellen

In het verleden heeft de doorbelasting op basis van de klassieke prestatie-eenheden CVE-tijd en geheugengebruik plaatsgevonden. In ASZ-termen: vups (VAX), gigabytes en mips (ICL). In 1996 is er een andere situatie ontstaan. ASZ werd een zelfstandige werkmaatschappij binnen de GAK-Holding en besloot om met ingang van 1 januari 1997 op een andere wijze van doorbelasten over te gaan. In plaats van te werken met prestatie-eenheden is, daar waar mogelijk, gekozen voor het doorbelasten op basis van BR-licenties. Voor een aantal applicaties was het (nog) niet mogelijk om op deze manier te werken.

Bij het doorbelasten van kosten wordt doorgaans onderscheid gemaakt tussen online- en batch-verwerking. Bij ASZ is geen sprake van een dergelijke consequente scheiding. In de wandelgangen zegt men dat batch-verwerking 's nachts plaatsvindt en niet wordt doorbelast. De feitelijke situatie is dat doorbelasting alleen plaatsvindt binnen een bepaald tijdsinterval, het dag-tijdsinterval of dag-window dat geldt van 07.00 tot 18.00 uur. Binnen dit dag-window worden alle prestatie-eenheden met betrekking tot zowel online-verwerking als batch-verwerking doorbelast. De registratieprogrammatuur is niet in staat om aan te geven of de geleverde prestatie-eenheden te maken hebben met batch-verwerking of met online-verwerking. Het uitlopen van een batch-job in de ochtenduren (van het nacht-window in het dag-window) betekent dat de prestatie-eenheden die geleverd worden binnen het dag-window worden doorbelast.

Het tarief van de BR-licentie

Uitgangspunt bij het vaststellen van de tarieven voor de BR-licentie was: de som van de kosten voor het gebruik van vups, gigabytes, mips en beheeruren dient te worden opgebracht door de som van de 'verkochte' licenties. Het tarief van de licentie is vastgesteld op basis van 'gemiddelde kosten', met daaraan toegevoegd een opslagpercentage/risicomarge. Op jaarbasis bedraagt de totale doorbelasting op basis van BR-licenties meer dan 10 miljoen gulden.

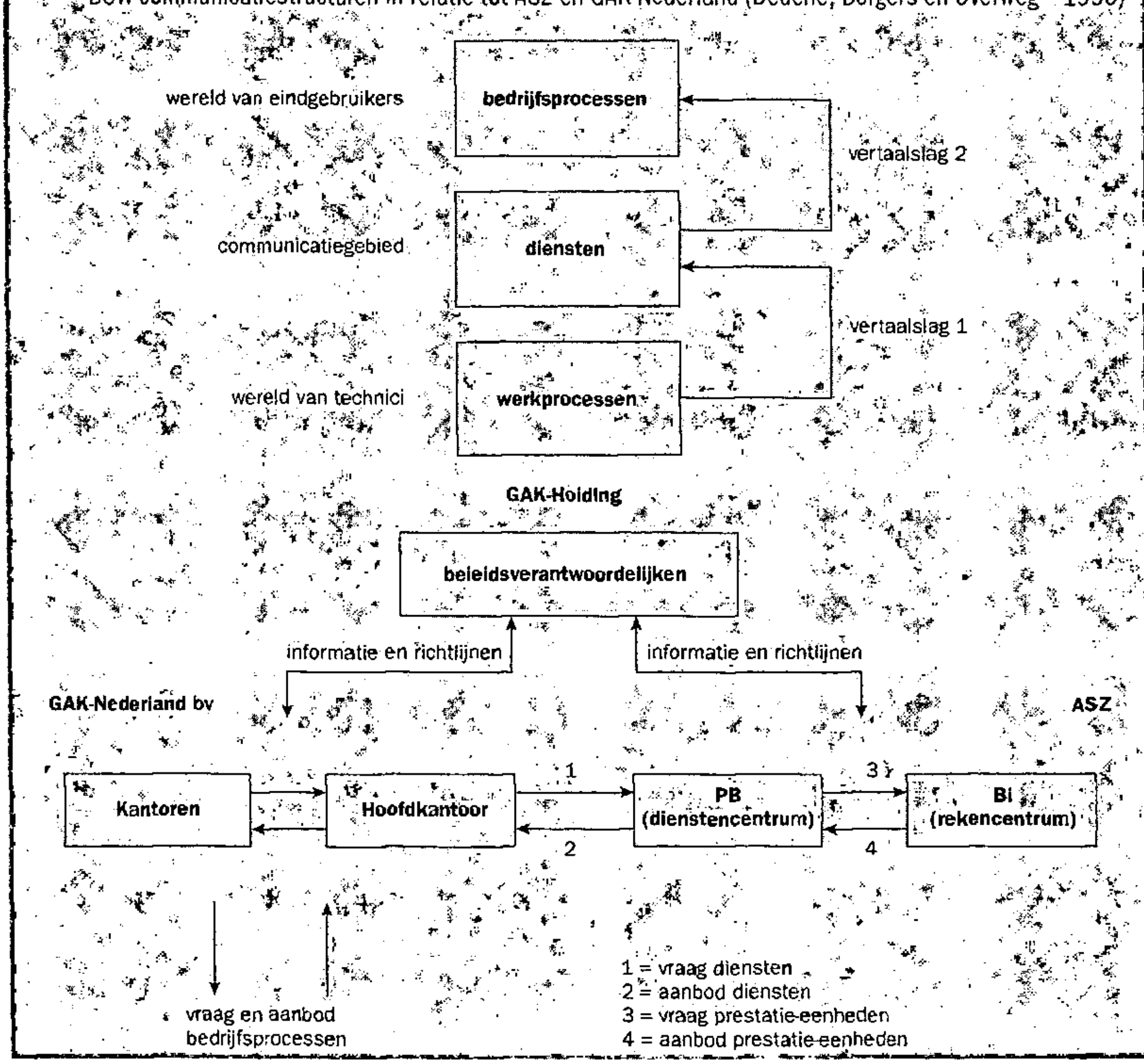
De omvang van de exploitatiekosten van applicaties die niet via BR-licenties kan worden verrekend is substantieel en kan op jaarbasis worden geschat op ongeveer 3 miljoen gulden.

Enkele activiteiten:
- de afdeling BI fungeert als een kostencentrum dat de 'rekencentrumfunctie' binnen ASZ vervult (zie figuren 1 en 3). De gemaakte kosten worden ondermeer bij het systeemhuis in rekening gebracht, waaronder PB;
- ASZ als geheel functioneert als een profitcentrum. Dit geldt ook voor het systeemhuis, dus ook PB moet commercieel werken (verkoopfunctie);
- de batch-verwerking wordt thans niet doorbelast. De omvang van het batch-werk en de benodigde inspanning van BI daarbij, neemt toe. Het is dan ook de vraag of wel of niet tot doorbelasting moet worden overgegaan;
- BI heeft contracten afgesloten met haar leveranciers om aan haar contractuele onderhoudsverplichtingen te kunnen voldoen ('underpinning contracts');
- PB sluit met BI 'underpinning contracts' af om aan haar verplichtingen jegens haar klanten te kunnen voldoen;
- BI bepaalt aan de hand van het aantal BR-licenties de benodigde capaciteit. Een belangrijk onderdeel van deze capaciteit is de zogenoemde 'Verwerkingskracht Interactief'. Het betreft de computercapaciteit die tijdens interactieve uren beschikbaar is. Gedurende de gehele periode is de beschikbare capaciteit gelijk. Door het interactieve karakter van het gebruik wordt de capaciteitsbehoefte gebaseerd op de verwachte piekperioden. Een en ander betekent dat de 'gereserveerde capaciteit' altijd moet worden betaald, ook als deze niet (volledig) gebruikt wordt;
- BI verzamelt 24 uur per dag alle gegevens betreffende de exploitatie en het gebruik van de informatiesystemen die tot de basisregistraties behoren en stelt deze

ter beschikking van PB en de afdeling financiële en economische zaken (FEZ). Om het verzamelen mogelijk te maken beschikt BI over een aantal gereedschappen, waaronder het TCS (Transactie Counting Systeem). De gereedschappen zijn eigen ontwikkelingen van ASZ. Ze maken gebruik van en zijn gebaseerd op de registratiegegevens die door de VAX-systemen worden opgeleverd. De meetgegevens worden op het niveau van 'usernames' zichtbaar gemaakt;

- FEZ stelt in overleg met PB de prijzen vast van de BR-licenties. Met behulp van de gegevens uit de rapportagesystemen en een opslagpercentage/risicomarge worden de prijzen van de licenties berekend en vastgesteld. Het aantal licenties bedraagt ongeveer 7 duizend;
- sommige teams verrichten diensten voor PB. De kosten (in beheeruren) worden in rekening gebracht;
- PB stelt BR-licenties ter beschikking van andere productteams en factureert de bijbehorende kosten. Deze belasten de in rekening gebrachte kosten via hun productenportfolio door aan de gebruikers;
- PB stelt licenties ter beschikking van de gebruikersorganisatie en factureert de bijbehorende kosten. Enkele maanden voor het einde van het jaar wordt het nieuwe aantal BR-licenties in overleg vastgesteld. Op basis van deze aantallen en het verbruik van bet totaal aantal BR-

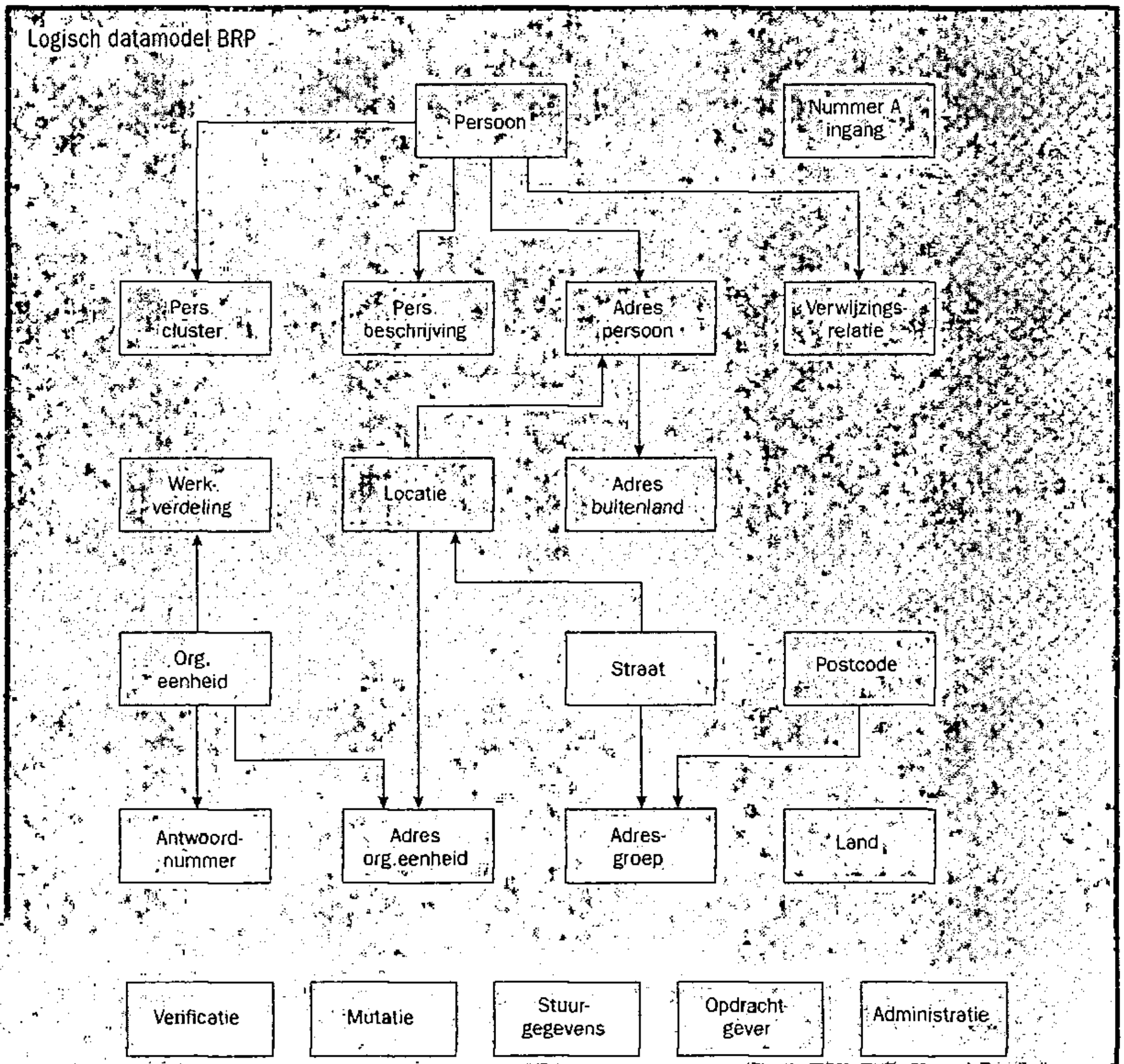
BSW-communicatiestructuren in relatie tot ASZ en GAK-Nederland (Dedene, Borgers en Overweg – 1990)



figuur 1

licenties in de afgelopen periode wordt (door FEZ in samenwerking met PB) de nieuwe prijs van de licenties voor het komende jaar bepaald.

De basisregistraties vormen de ruggengraat van de informatievoorziening binnen het GAK. Vanaf 14 duizend werkplekken wordt dagelijks met de systemen van ASZ gewerkt. Deze systemen maken gebruik van de basisregistraties. De kosten van deze activiteiten (raadplegingen, mutaties, online/batch-georiënteerd) komen voor een deel ten laste van de gebruikersorganisatie op het hoofdkantoor, de afdeling Basisregistraties (BRS). Deze afdeling krijgt een deel van de totale kosten doorbelast. Ook worden de districtskantoren via BR-licenties doorbelast. BRS heeft daardoor geen goed inzicht in de totale kosten. Bovendien wil BRS een duidelijke relatie zien tussen de kosten en de door ASZ, op verzoek van BRS, geleverde prestaties. BRS heeft de indruk dat, hoewel het aantal handelingen door de gebruikers niet stijgt, de kosten wel toenemen. BRS wil dat alle kosten met betrekking tot het gebruik en beheer van de basisregistraties bij BRS in rekening worden gebracht. Bovendien moet PB informatie leveren, opdat BRS binnen het GAK de gemaakte kosten kan doorbelasten.



figuur 2

3.3 De basisregistraties

In deze paragraaf wordt ingegaan op de basisregistratie personen (BRP). De basisregistraties van werkgevers (BRWG) en dienstverbanden (BRDV) worden in het kader van dit artikel buiten beschouwing gelaten, maar kunnen op dezelfde wijze worden beschreven.

Omdat bij doorbelasting kwantiteiten een belangrijke rol spelen, worden naast de logische structuur van de database en subsystemen en -processen ook kwantitatieve gegevens gegeven. Daarbij is niet gestreefd naar volledigheid, maar naar een sfeertekening die een belangrijke rol kan spelen bij de keuze van de gewenste oplossingsrichting.

Subsystemen en -processen

De BRP bevat de persoonsgegevens van alle verzekerden waarmee het GAK, ten behoeve van de administratie van haar opdrachtgevers, relaties onderhoudt. Deze basisregistratie bestaat uit 51 processen die in de volgende vier subsystemen zijn gegroepeerd:
- registreren persoonsgegevens;
- registreren verwijsgegevens;
- onderhouden verificatiestatus persoonsgegevens;
- registreren niet-persoonsgebonden gegevens.

Het logisch datamodel kan als in figuur 2 worden weergegeven.

De totale omvang van de database is 11 gigabytes en bevat onder andere de volgende record-types met de daarbij behorende aantallen record-occurrences:

Recordtype	Aantal occurrences
Persoon	9 miljoen
Persoonsbeschrijving	15 miljoen
Persoonscluster	11 miljoen
Verwijzingsrelatie	10 miljoen
Opdrachtgever	150
Org. Eenheid	1.700
Adres buitenland	400 duizend
Land	600
Adres persoon	15 miljoen
Locatie	6 miljoen
Straat	230 duizend
Adresgroep	600 duizend
Postcode	500 duizend

Gebruik

Met behulp van onderstaande gegevens wordt een globaal beeld gegeven van het gebruik van de BRP-database. Er is niet gestreefd naar volledigheid, maar de gegevens zijn bedoeld voor de discussie over de nieuwe wijze van doorbelasting.

activiteiten/ diensten/ producten	hoeveelheid
raadplegen regio-kopieën, online en batch	3.500 duizend per week
raadplegen centrale-kopie, online en batch	600 duizend per week
raadplegen centrale database, batch	60 duizend per week
bulk-raadplegingen	8 miljoen per jaar
muteren centrale database	225 duizend per week
extract mutaties	dagelijks

Transacties

Het Project Aansluiting Deeladministraties (PAD) op de BRP voorziet in een mechanisme waar deeladministraties gebruik van kunnen maken om gegevens op te vragen of te muteren in de BRP. Het betreft zowel batchverwerking als online-verwerking. Het aantal beschikbare transacties dat bij het opvragen en muteren van de BRP een rol speelt, bedraagt ruim 40. Enkele voorbeelden van transacties zijn het opvragen of wijzigen van:
- persoonsgegevens;
- land;
- organisatorische eenheid;
- adres.

4 Gewenste situatie

PB wil komen tot een adequate (effectieve) doorbelasting van de kosten van het beschikbaar stellen (exploitatie, onderhoud en gebruik) van de basisregistraties aan afnemers. Dat betekent dat voldaan moet worden aan de volgende doelstellingen:
- in de gewenste situatie mogen de in hoofdstuk 2 genoemde knelpunten niet meer voorkomen en moet voldaan zijn aan de aanvullende eisen;
- PB fungeert als profitcentrum en levert, tegen gerechtvaardigde tarieven, diensten aan met name het GAK, maar ook aan andere afnemers.

4.1 Eisen te stellen aan de gewenste situatie

Wijze van doorbelasting

Het GAK-onderzoek van Gleusteen/Huygen geeft onder andere als conclusie: 'Prestatie-eenheden (cpu-seconden, I/O-opdrachten, geheugengebruik) zijn nietszeggend voor de gebruiker'.

Ook in het tactische ITIL-proces 'Cost Management' en in recente literatuur wordt aangegeven dat het noodzakelijk is om bij doorbelasting te streven naar voor afnemers begrijpelijke factureerbare eenheden. Dit sluit aan bij het gesignaleerde knelpunt in hoofdstuk 2. Daarom moet doorbelasting bij voorkeur plaatsvinden op basis van gebruikerseenheden die in overleg met de gebruikersorganisatie worden vastgesteld. Deze gebruikerseenheden moeten voor de gebruiker herkenbaar zijn, een goede maatstaf vormen voor het gebruik van de basisregistraties en aansluiten bij de organisatiecultuur en de wijze van doorbelasten bij het GAK.

PB sluit overeenkomsten met BI (Beheer Infrastructuur) op basis van een klant/leverancierrelatie. Deze overeenkomsten zijn voor PB de 'underpinning contracts'. BI moet naast inzicht in de kosten, ook informatie opleveren waarmee PB de afnemers kan voorzien van de juiste aantallen/omvang van de gebruikerseenheden. Deze werkwijze sluit ook aan bij de zogenaamde 'verzelfstandigingsmethode'. Deze methode betekent dat iedere gebruiker zelf verantwoordelijk is voor (de kosten van) zijn automatisering en dat elke afdeling wordt gezien als een 'zelfstandig bedrijf' binnen een bedrijf (zie figuur 3). Hiermee wordt de afdelingschef verantwoordelijk voor het IT-gebruik.

Eisen aan een tariefstelsel

Het gehanteerde tariefstelsel moet voldoen aan de algemene eisen die gelden voor tariefstelsels: effectief, eenvoudig, herkenbaar, robuust, rechtvaardig en beïnvloedbaar. Bovendien mogen de risico's voor PB slechts beperkt zijn. Het structureel aanpassen van het verrekeningssysteem mag, vanuit beleidsoogpunt, zo min mogelijk gebeuren. Het tariefstelsel moet daarom 'toekomstvast' zijn.

4.2 Toekomstige ontwikkelingen

Het GAK is veruit de belangrijkste klant van ASZ. Een tariefstelsel moet derhalve in ieder geval goed passen in de zakelijke relatie met het GAK. Dat neemt niet weg dat ASZ ook andere klanten heeft. Bovendien zijn er ontwikkelingen (samenwerkingsverband met bijvoorbeeld Achmea) die kunnen leiden tot een sterkere marktpositie van ASZ.

5 Oplossingsmogelijkheden

Doorbelasting kan op verschillende niveaus plaatsvinden. Het BSW-communicatiemodel onderscheidt drie niveaus van werklast: de werkprocessen, de diensten en de bedrijfsprocessen. In figuur 1 is het model toegepast op ASZ en het GAK.

Werkprocessen

Werkprocessen zijn de activiteiten die de IT-dienstenorganisatie (bijvoorbeeld ASZ) betreffen. Hier zijn de werklasttermen vups, mips en gigabytes van toepassing. Doorbelasting op basis van deze prestatie-eenheden voldoet niet aan de gestelde eisen en kan daarom niet worden toegepast. Doorbelasting moet derhalve op basis van bedrijfsprocessen of diensten plaatsvinden.

Diensten

Tussen de bedrijfsprocessen en werkprocessen liggen de diensten die de IT-organisatie aan de afnemers levert. Voorbeelden zijn: muteren van het adres van een persoon of het verwijderen van een persoon uit de BRP-database. Het bepalen van de kosten van diensten kan worden gerealiseerd door van iedere dienst de bijbehorende werkprocessen te bepalen (vertaalslag 1, zie figuur 1).

Bedrijfsprocessen

Met bedrijfsprocessen wordt de realisatie van bedrijfsdoelen nagestreefd. Op dit niveau drukt men de werklast uit in bedrijfstermen zoals *het toevoegen persoon*, *het muteren van een dienstverband*, *het berekenen van een uitkering*. Zo omvat het bedrijfsproces *toevoegen persoon* een aantal diensten als het on-line inbrengen van een aantal persoonsgegevens en het controleren van de betreffende gegevens bij een externe organisatie (bijvoorbeeld de Gemeentelijke Basisadministratie of de fiscus). Het bepalen van de kosten van bedrijfsprocessen kan worden gerealiseerd door van ieder bedrijfsproces de bijbehorende diensten te bepalen (vertaalslag 2, zie figuur 1).

Een dergelijke vertaalslag moet voor ieder bedrijfsproces worden gemaakt. Dit betekent dat ook voor ieder informatiesysteem een vertaalslag moet worden gemaakt. Dat is een omvangrijk werk. Bovendien veranderen informatiesystemen regelmatig en neemt het aantal informatiesystemen bij ASZ en haar klanten toe.

De dienstverlening van PB is: 'Het beschikbaar stellen van operationele applicaties en gegevens met betrekking tot de basisregistraties'.

Het hoeft geen probleem te zijn om van de door PB geleverde applicaties een vertaalslag te maken. De basisgegevens worden echter ook aan andere productteams, die de gegevens met behulp van eigen applicaties benaderen en bewerken, beschikbaar gesteld. Ook voor deze informatiesystemen moeten vertaalslagen worden gemaakt en het gebruik moet op dat niveau worden geregistreerd. Dat valt buiten de taak van PB. Sterker nog, PB kan daarvoor geen verantwoordelijkheid dragen.

5.1 Oplossingsrichting

Op grond van het bovenstaande verdient het de voorkeur om de doorbelasting op de bron te richten: de diensten die worden verleend op de gegevensverzamelingen personen (BRP), werkgevers (BRWG) en dienstverbanden (BRDV). Door het gebruik en beheer bij de bron te meten en in voor de gebruikers begrijpelijke eenheden door te belasten, wordt het beste aan de gewenste situatie voldaan. (Men kan de diensten met een verzameling legoblokjes vergelijken, van diverse kleuren, formaten en prijzen. Voor de klanten zijn de werkprocessen, in casu de grondstoffen waaruit legoblokjes zijn gemaakt, niet interessant. Wel interessant zijn de bedrijfsprocessen, datgene wat de klant doet met de legoblokjes).

6 Uitwerking van de oplossingsrichting

Bij het uitwerken van de oplossingsrichting wordt gebruik gemaakt van de volgende stappen naar een tariefstelsel:

1 doelstelling, randvoorwaarde en omgevingsfactoren;
2 definitie IT-diensten;
3 kostenverdeelmodel;
4 opzet tariefstelsel;
5 ontwerp en implementatie tariefomgeving;
6 evaluatie tariefstelsel.

6.1 Doelstelling, randvoorwaarde en omgevingsfactoren

Doelstelling

Doel van het onderzoek is om ten behoeve van het domein van de drie basisregistraties tot een wijze van doorbelasten te komen van het beheer (exploitatie en onderhoud) en gebruik van de gegevensverzamelingen en applicaties die beboren tot de drie basisregistraties, die:
- beter aansluit bij de bedrijfsprocessen van de afnemers;
- beter herkenbaar is;
- de mogelijkheid vergroot om de omvang van de in rekening gebrachte kosten te beïnvloeden;
- leidt tot beheersing van en (bij gelijk verbruik) verlaging van de in rekening gebrachte kosten.

Randvoorwaarde

In hoofdstuk 5 is geconcludeerd dat het de voorkeur verdient om de doorbelasting op de diensten te richten die op de gegevensverzamelingen BRP, BRWG en BRDV worden verleend. Het gebruik en beheer van deze databases moeten bij de bron (toegang tot de databases) worden gemeten en in voor gebruikers begrijpelijke eenheden worden doorbelast.

Omgevingsfactoren

Het ASZ-businessplan 'Visie en strategie 1996-2000' bevat een groot aantal omgevingsfactoren, zoals:
- ASZ committeert zich aan het marktconform leveren van moderne oplossingen;
- exploitatie en beheer moeten aan (performance-)eisen voldoen die tot een grotere klanttevredenheid bij het GAK leiden;
- een scherper inzicht in en beheersing van de kosten is noodzakelijk;
- een verdere verhoging en met name het waarborgen van de kwaliteit van de dagelijkse exploitatie staan hoog op de agenda van het ASZ-management;
- ASZ moet zich ontwikkelen tot een proces-georiënteerde organisatie;
- ASZ zal (h)erkend worden als business- en systemintegrator, die bijdraagt aan de effectiviteit, gebruikerstevredenheid en kostenbeheersing;
- bij een onveranderd takenpakket (en afgezien van ontwikkelingen in lonen) zal ASZ ernaar streven om in het jaar 2000 een kostenniveau te bereiken dat ten opzichte van 1996 gemiddeld 23 procent lager ligt.

6.2 Definitie IT-diensten

De houders (gedelegeerde/gemandateerde eigenaren) van de drie basisregistraties hebben hun gegevens in bewaring gegeven bij PB. Dit productteam bewaart en koestert de gegevens en stelt deze op elk gewenst moment beschikbaar aan gebruikers die bevoegd zijn de gegevens te gebruiken. Deze gebruikers zijn daartoe door of namens de houder geautoriseerd.

PB heeft derhalve met beheer- en gebruiksdiensten te maken. Met beide diensten zijn kosten gemoeid. Deze worden doorgaans door BI en/of PB gemaakt, maar ook andere partijen kunnen deze bij PB of BI in rekening brengen.

In figuur 3 zijn de zelfstandige partijen weergegeven die een rol spelen in de doorbelastingsketen. Ter toelichting een voorbeeld van 'underpinning aspects'. Indien PB met het GAK afspraken maakt over de snelheid waarmee een probleem aangepakt of opgelost moet worden, dan zal PB met BI dusdanige afspraken moeten maken dat BI tenminste hetzelfde niveau van dienstverlening aan PB verleent. BI zal op zijn beurt harde afspraken maken met externe leveranciers. Als BI met een netwerkleverancier een contract heeft lopen waarin staat dat bij een netwerkstoring binnen 4 uur wordt begonnen met het oplossen van het probleem, dan kan PB met het GAK geen afspraak maken dat het probleem binnen 2 uur wordt aangepakt.

De beheerdiensten

Zowel door BI als PB worden beheerdiensten verricht. BI verricht de bekende activiteiten ten behoeve van het beheer (exploitatie en onderhoud) zoals:

- beschikbaar stellen van schijfruimte, verwerkingscapaciteit;
- bewaken van de vullingsgraad van de schijfruimte;
- maken van backups;
- inrichten en uitvoeren van restore-voorzieningen;
- 24-uur bewaking/stand by;
- bewaken performance (tuning);
- eerste- en tweedelijnsondersteuning;
- beheren toegangsbeveiliging.

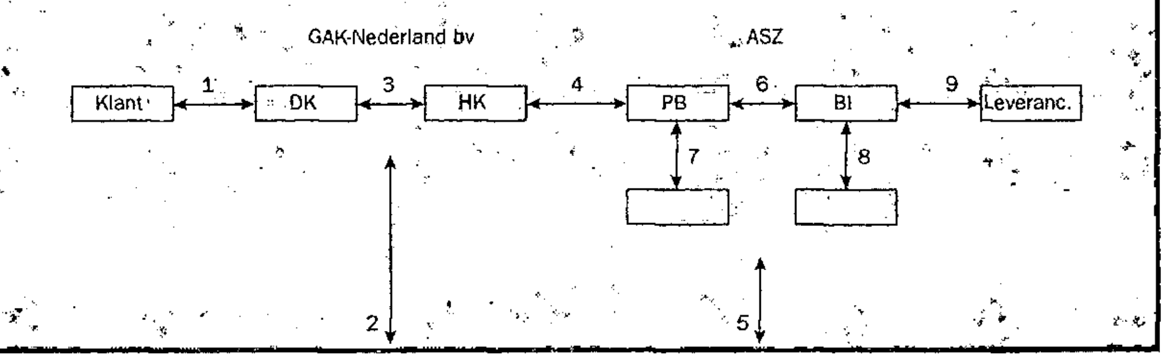
Binnen PB is de groep *SUPPORT* belast met het beheer van de basisregistraties en deze verricht onder andere de volgende activiteiten:

- draaien/verzorgen batch-productie;
- verzorgen van de helpdeskfunctie;
- intake releases;
- bepalen/bewaken batch-beschikbaarstellingstijden;
- vervullen van de informatiefunctie.

Verklaring van de relaties in figuur 3:

1 De klanten van de districtskantoren leveren en/of krijgen informatie.
2 Het GAK fungeert binnen het veld van de sociale zekerheid als informatieleverancier, maar ontvangt ook informatie vanuit het veld.
3 De districtskantoren worden door het hoofdkantoor voorzien van informatie en worden door het hoofdkantoor doorbelast voor het gebruik van de basisgegevens.
4 Het GAK vraagt en krijgt diensten betreffende beheer en gebruik van de basisgegevens. Daartoe worden door het GAK en PB afspraken gemaakt (service level management).
5 ASZ is een commercieel centrum voor informatieverwerking en biedt haar diensten aan binnen het veld van de sociale zekerheid.
6 PB vraagt en krijgt bewaardiensten en prestatie-eenhe-

Zelfstandige partijen in de doorbelastingsketen met de relatie GAK en PB als centrum



figuur 3

den van Beheer Infrastructuur. PB sluit daartoe overeenkomsten af met BI. Deze overeenkomsten zijn voor PB als het ware de 'underpinning contracts'.

7 PB vraagt en krijgt diensten van afdelingen binnen ASZ (zoals bijvoorbeeld kantoorautomatisering).
8 BI vraagt en krijgt diensten van afdelingen binnen ASZ.
9 BI zorgt voor 'underpinning contracts' met haar externe leveranciers.

De gebruiksdiensten

In hoofdstuk 5 is geconcludeerd dat het de voorkeur verdient om de doorbelasting op de bron te richten: de diensten die op de gegevensverzamelingen BRP, BRWG en BRDV worden verleend. Het gaat daarbij om doorbelastingseenheden die voor gebruikers/afnemers begrijpelijk zijn.

In paragraaf 3.3 is een opsomming gegeven van de subsystemen/processen/activiteiten die werken op de BRP. Ook zijn enkele op de BRP gedefinieerde transacties genoemd, waarmee zowel online als in batch de database kan worden geraadpleegd en gemuteerd.

Bij de doorbelasting van de gebruiksdiensten staan in principe twee wegen open:

- doorbelasting op basis van het gebruik van functionele processen/activiteiten;
- doorbelasting op basis van het gebruik van functionele transactiecodes.

Daarbij doen zich enkele problemen voor:

- het aantal functionele processen/activiteiten is omvangrijk (enkele honderden);
- het aantal soorten functionele transacties is tamelijk omvangrijk (ongeveer 100);
- de verzameling transacties is recentelijk in verband met de ontwikkeling van een nieuwe applicatie substantieel uitgebreid. Het is dus geen vaste verzameling. Overlap in transacties komt voor (meerdere transacties die geheel of gedeeltelijk hetzelfde doen);
- de relatie tussen een functioneel proces/activiteit en één of meer transacties is niet altijd duidelijk;
- er zijn (online-)processen die de databases buiten de transacties om benaderen.

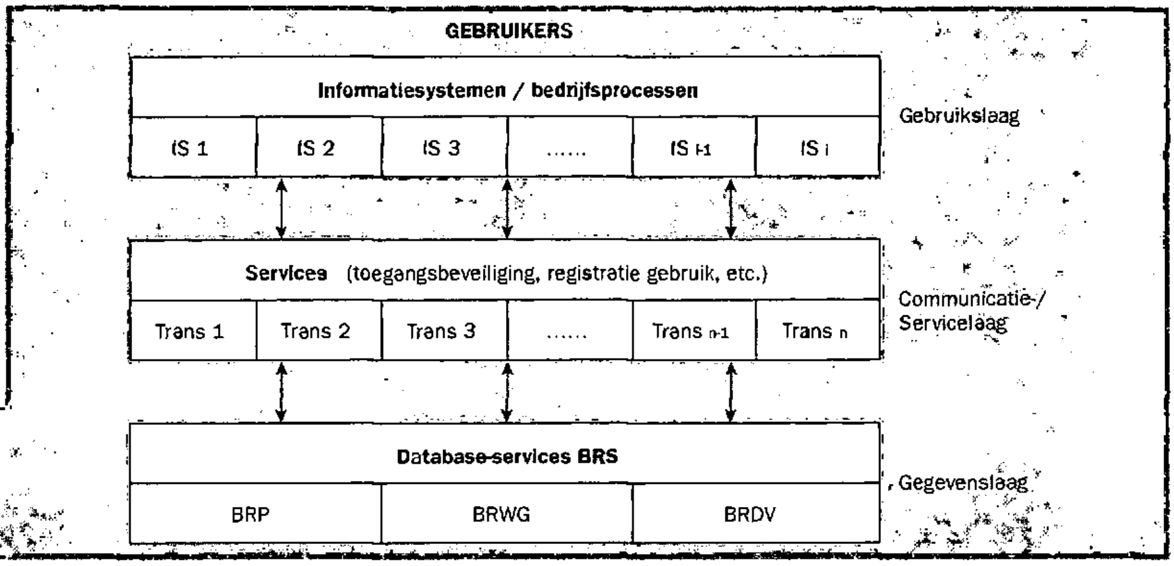
Er moet een keus worden gemaakt welke van de twee manieren van doorbelasting de voorkeur verdient. Indien gekozen wordt voor functionele processen/activiteiten dan moet:

- het aantal functionele activiteiten, met behulp waarvan wordt doorbelast, worden beperkt. Het is dan zinvol om tot geclusterde functionele activiteiten te komen of tot een representatieve set functionele activiteiten. De keuze moet op iteratieve wijze door ASZ en het GAK worden vastgesteld;
- het gebruik van de functionele activiteiten moet geregistreerd worden. Dit betekent dat het gebruik van functionele activiteiten in de betreffende informatiesystemen wordt geregistreerd of dat er een duidelijke relatie met transactiecodes kan worden gelegd.

Alvorens tot een keuze te komen, is het zinvol om op een conceptuele manier de problematiek te benaderen en vooral ook naar de toekomst te kijken. In verband met de rol van de basisregistraties kan het gebruik van de basisgegevens met een drielagenmodel worden weergegeven. Het betreft de volgende lagen (zie figuur 4):

- gebruikslaag/bedrijfslaag;
- communicatielaag/servicelaag;
- gegevenslaag.

Het model doet denken aan het BSW-model, maar heeft toch een wat andere insteek. Uitgangspunt is dat een groot aantal informatiesystemen communiceert met de basisgegevens. Uit het oogpunt van het beheer is het ongewenst dat individuele informatiesystemen rechtstreeks op de databases werken. Individuele programmeurs kunnen dan op hun eigen manier de databases bespelen en/of bewerken, met alle mogelijke gevolgen voor andere gebruikers. Daarom dient de beheerder van de databases over een aantal services te beschikken, die hij aan een ieder kan aanbieden die de basisgegevens wenst te benaderen en te bewerken (en daartoe geautoriseerd is). Deze services zitten in de tussenlaag, de servicelaag/communicatielaag. De data(base)services kunnen bestaan uit het ophalen, wijzigen, toevoegen of verwijderen van één of meer record-occurrences. De communicatieservices kunnen bijvoorbeeld uit toegangsbeveiliging en/of registratie van het gebruik van de databases bestaan. Een groot voordeel van deze benadering is dat bij vervanging van de databases door nieuwe die bijvoorbeeld door een ander DBMS worden ondersteund, de informatiesystemen niet gewijzigd hoeven te worden. Door het aanpassen van de service-modules hoeven de gebruikers er in feite niets van te merken. Als de services naast een technische realisatie een functionele betekenis hebben, dan kan de doorbelasting eveneens in functionele termen plaatsvinden.



figuur 4

De huidige transactiecodes vormen een functionele service (opvragen of wijzigen van één of meer record-occurrences). Een deel van de transactiecodes (een basispakket) wordt op initiatief van PB aangeboden. Andere zijn gemaakt op verzoek van of namens gebruikers, die ze integreren in hun applicatie-omgeving. De codes voldoen goed aan de functionele service-elementen uit de bovengenoemde tussenlaag. Het doorbelasten op basis van transactiecodes geeft aan de gebruikers overzicht over het gebruik van de databases. Geconcludeerd kan worden dat de transacties, in combinatie met het drielagenmodel, in principe een uitstekende basis vormen voor het doorbelasten van het gebruik van de databases. Belangrijk is dat deze werkwijze bovendien 'toekomstvast' is.

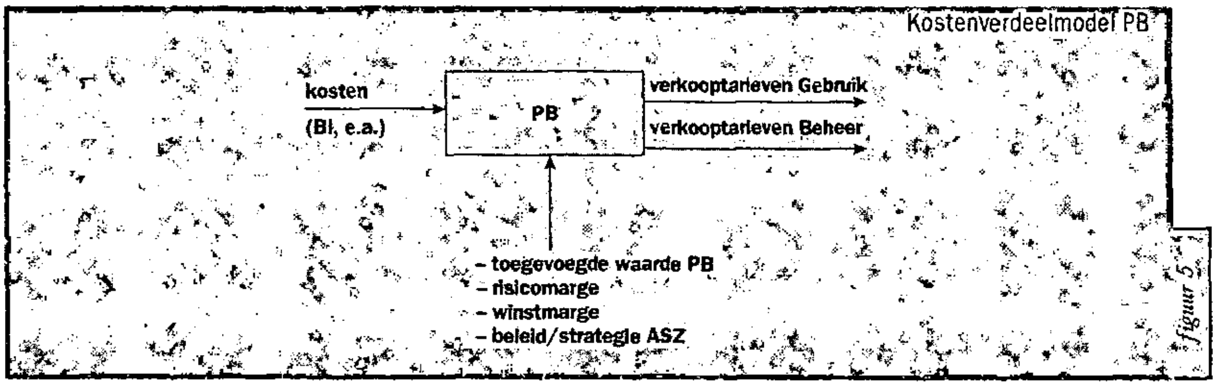
Er moet wel een tijdelijke oplossing gevonden worden voor het doorbelasten van het 'rechtstreekse' gebruik van de databases. Het is gewenst dat dergelijk gebruik wordt afgebouwd. Een belangrijk voordeel is dat de overgang naar het doorbelasten op basis van gebruikte transacties vergemakkelijkt wordt, aangezien het gebruik al met behulp van het Transactie Counting Systeem (TCS) wordt geregistreerd.

6.3 Kostenverdeelmodel

Het kostenverdeelmodel bevat de regels die uiteindelijk de kostprijzen voor de tarifering leveren. Vanuit PB gezien zijn de belangrijkste kosten:
- de door BI (het rekencentrum van ASZ) in rekening gebrachte kosten voor gebruik en beheer van de basisregistraties;
- de door PB geleverde toegevoegde waarde;
- de overige kosten die door andere ASZ-onderdelen in rekening worden gebracht.

De totale kosten, met de daaraan toegevoegd risicofactoren, winstfactoren en beleidsaspecten van ASZ, vormen de basis voor de tariferingsregels (de verkoopsom, zie figuur 5). Een deel van de verkoopsom moet worden opgebracht door tarifering van de beheer- en bewaarfuncties (de verkoopsom Beheer). Het resterende deel moet door doorbelasting van het gebruik van de databases (de verkoopsom Gebruik) worden opgebracht. Over deze verdeling moet een afspraak worden gemaakt.

PB heeft besloten de verkoopsom Gebruik te splitsen in een deel voor *raadpleegfuncties* met eenvoudige online-mutaties en een deel voor het (massaal) *bijwerken* van de gegevens, voornamelijk via batch-verwerking. Dit betekent dat er met de volgende drie verkoopsommen wordt gewerkt:
- *Beheer*: exploitatie en onderhoud;
- *Gegevensgebruik*: het raadplegen van de gegevens en het aanbrengen van eenvoudige mutaties door mede-



Kostenverdeelmodel PB
figuur 5

werkers die daartoe bevoegd zijn. Dit raadplegen kan direct, medewerkers kunnen een expliciete opdracht aan specifieke informatiesystemen geven, of indirect via informatiesystemen die de gegevens nodig hebben. Bij het raadplegen en muteren wordt gebruik gemaakt van functionele transacties. Een uitzondering hierop vormt de levering van database-extracten en mutatiebestanden (batch-verwerking);

- *Gegevensonderhoud*: het bijwerken (updaten) van de gegevens door medewerkers die daartoe bevoegd zijn. Dit bijwerken kan direct, medewerkers kunnen een expliciete opdracht geven aan specifieke informatiesystemen, of indirect via informatiesystemen die de gegevens via batch-verwerking bijwerken. De programmatuur voor gegevensonderhoud werkt (nog) niet met functionele transacties.

Bij het verdelen van de verkoopsom van 1998 over de drie rubrieken spelen de volgende twee criteria een rol:

- kosten die aan de betreffende verkooppost kunnen worden toegerekend;
- beleidsmatige aspecten.

Tabel 1 geeft een overzicht van de drie verkoopsommen. In dit artikel wordt de doorbelasting van het gegevensgebruik verder uitgewerkt.

6.4 Opzet tariefstelsel gegevensgebruik

Eisen aan een tariefstelsel

Hoewel een tariefstelsel voor iedere organisatie uniek is, gelden als algemene kwaliteitseisen: effectiviteit, eenvoud, herkenbaarheid, robuustheid, rechtvaardigheid, beïnvloedbaarheid en 'toekomstvastheid'. Een tariefstelsel dat gebaseerd is op het doorbelasten van het gebruik van functionele transacties, voldoet aan deze kwaliteitseisen.

De verkoopsom Gebruik moet grotendeels aan de hand van de gebruikte transacties worden doorbelast. Een uitzondering vormt (voorlopig nog) de aan afnemers geleverde database-extracten en mutatiebestanden. De betreffende programmatuur werkt deels (nog) niet met functionele transacties. Om toch tot een doorbelasting op basis van functionele eenheden te komen, kan voor een prijs per geleverd record worden gekozen of een prijs per

Tabel 1 Overzicht verkoopsom(men) 1998

Verkoopsom 1998	Bedrag	Bedrag
Beheer (I)		f 3.000.000
Gegevensgebruik (II)		f 10.000.000
a functionele transacties	f 9.000.000	
b leveren van database-extracten	f 500.000	
c leveren van mutatiebestanden	f 500.000	
Geg. onderhoud (III)		f 3.000.000
Totaal		f 16.000.000

geleverd teken óf een combinatie van deze twee mogelijkheden. Ook kan op basis van specifieke afspraken in Service Level Agreements (SLA's) worden doorbelast. ASZ berekent op basis van het aantal records. De prijs per record wordt eenvoudig vastgesteld: miljoen gulden gedeeld door het aantal geleverde records in 1997.

Functionele transacties

Hoe kan het gebruik van functionele transacties worden doorbelast? Daartoe is het nodig de zwaarte van de transacties te kennen. Dit kan worden gerealiseerd door precies vast te leggen welke prestatie-eenheden bij een transactie behoren. Dit levert echter problemen op. Een betere methode is om een schatting te maken van de relatieve zwaarte van de transacties. Door aan iedere transactie een gewichtsfactor (bijvoorbeeld tussen 1 en 100) toe te kennen, kan de prijs van een transactie-eenheid en vervolgens van de transacties worden vastgesteld. *De prijs van de transactie-eenheid = de geschatte verkoopsom Gebruik functionele transacties (zie tabel 1) / (som van alle transacties: jaarlijks gebruik van een transactie*gewichtsfactor).* Om de prijs van de transactie-eenheid te kunnen vaststellen zijn de volgende gegevens nodig:

- de verkoopsom Gegevensgebruik functionele transacties (zie tabel 1);
- de gebruiksfrequenties van de functionele transacties;
- de gewichtsfactoren van de functionele transacties.

Gebruiksfrequenties

Het gebruik van functionele transacties wordt geregistreerd met behulp van het Transactie Counting Systeem (TCS). De gebruiksfrequenties van 1997 zijn opgenomen in tabel 2.

Gewichtsfactoren

Functionele transacties verschillen in zwaarte. Een eenvoudige opvraagtransactie is aanzienlijk minder zwaar dan een pittige mutatietransactie. Er zijn meerdere manieren om transacties te wegen. De gewichtsfactor kan onder andere gebaseerd zijn op:

a het verbruik van prestatie-eenheden. Het doorbelasten op basis van prestatie-eenheden past niet binnen de gewenste oplossingsrichting;

b het aantal geraadpleegde en gemuteerde record-occurrences uit de logische database. Daartoe moet van iedere transactie worden vastgesteld wat het (gemiddelde) aantal geraadpleegde en gemuteerde record-occurrences is. Het lijkt een zeer omvangrijke taak om van ruim honderd functionele transacties het gemiddelde aantal opvraag-occurrences en het gemiddelde aantal gemuteerde record-occurrences weer te geven, maar door inschakeling van de juiste personen is dat echter in enkele dagen gerealiseerd. Vervolgens worden beide gecombineerd tot één gewichtsfactor. Daarbij is een mutatie viermaal zo zwaar beoordeeld als een opvraging;

c door de gebruikersorganisatie vast te stellen gewichten. Deze organisatie heeft dan de mogelijkheid beleid te voeren (te sturen) met betrekking tot het gebruik van de functionele transacties;

d door PB vast te stellen gewichten (beïnvloedingsfactoren). Daarbij kan onder andere aan een verschillende gewichtsfactor voor online-verwerking en batch-verwerking worden gedacht. Hoewel de betreffende functionele batch-transactie nagenoeg een kopie zal zijn van de functionele online-transactie, kan PB verschillende gewichtsfactoren toekennen;

e een combinatie van b en c;

f een combinatie van b en d.

In de gevallen e en f is er als het ware sprake van 'een gewichtsfactor over gewichtsfactoren'. Bij ASZ is voorlopig gekozen voor optie b. De gewichtsfactoren zijn opgenomen in tabel 2. Verwacht mag worden dat in de toekomst een combinatie van b en d de voorkeur zal genieten. Uit tabel 2 kan worden afgelezen dat de gemiddelde prijs van een functionele transactie drie cent bedraagt.

Opmerkingen

Het is belangrijk op te merken dat, indien het mogelijk is de relatie functionele transactie versus prestatie-eenheden te meten, het zinvol is deze meting steekproefsgewijs te doen plaatsvinden. Met behulp hiervan kan het plausibiliteitsgehalte van de volgens methode b gemaakte schattingen worden onderzocht. Indien er een groot verschil tussen de geschatte functionaliteit (aantal raadplegingen/mutaties) en de gemeten prestatie-eenheden bestaat, zijn er twee mogelijkheden:

- de geschatte functionaliteit is onjuist;
- de geschatte functionaliteit is juist. Dit betekent hoogstwaarschijnlijk dat de betreffende transactie zeer inefficiënt is geprogrammeerd. Nader onderzoek kan tot een vernieuwing/aanpassing van de betreffende transactie en besparing van kosten leiden. Dit heeft lagere tarieven voor de opdrachtgevers tot gevolg.

In tabel 2 is voor iedere functionele transactie één regel

Tabel 2 Rekenmodel tarieven functionele transacties

Functionele transacties	gewichtsfactor	aantal transacties 1997	product	prijs
BRP001	14	13.800.000	193.200.000	
BRP002	17	1.600.000	27.200.000	
BRP003	48	400.000	19.200.000	
BRP004	4	1.200.000	4.800.000	
BRP005	1	1.800.000	1.800.000	
BRP007	5	750.000	3.750.000	
BRP012	50	540.000	27.000.000	
BRP016	3	2.300.000	6.900.000	
...				
Tot. BRP		222.000.000		
BRWG001	32	340.000	10.880.000	
BRWG017	4	75.000	300.000	
BRWG020	22	4.700.000	103.400.000	
BRWG022	65	1.000	65.000	
BRWG027	12	3.400.000	40.800.000	
...				
Tot. BRWG		62.000.000		
BRDV001	28	1.340.000	37.520.000	
BRDV004	46	1.325.000	60.950.000	
BRDV058	10	2.510.000	25.100.000	
BRDV059	4	3.190.000	12.760.000	
...				
Tot. BRDV		16.000.000		
Alg. totaal		300.000.000		f 9.000.000

opgenomen. Indien het tijdstip waarop de transactie wordt uitgevoerd van belang is voor de tarifering, dan kunnen meerdere regels worden opgenomen. Bijvoorbeeld vier regels als een indeling naar nacht, ochtend, middag en avond gewenst is. Desgewenst kan ook voor ieder uur een regel worden opgenomen. Met behulp van de gewichtsfactor PB kan worden aangegeven welke vermenigvuldigingsfactor van toepassing is (gewichtsfactor: b*d), bijvoorbeeld: nacht =1, ochtend =3, middag=3 en avond =2. Dit betekent dat een transactie in de ochtend 3 keer zo duur is als dezelfde transactie in de nacht.

In tabel 2 zijn alle functionele transacties opgenomen. Een verdere verfijning kan door met drie afzonderlijke tabellen te werken, zodat met iedere corporate database van de basisregistraties een eigen beleid gevoerd kan worden.

Risicofactoren

De tarieven voor 1998 zijn gebaseerd op het gebruik van 1997. Dit gebruik is gecorrigeerd omdat substantiële veranderingen in 1998 verwacht worden. Een risicofactor is een significant ander gebruik van de diensten dan was verwacht. Om dit risico te beteugelen moeten de volgende maatregelen worden getroffen:

- de afspraken over afnamevolumes en prijzen moeten onderdeel uitmaken van een adequaat service level management. Daarin moet onder andere worden vastgelegd dat de tarieven gelden zolang het verwachte gebruik binnen zekere grenzen valt (bandbreedte, bij-

voorbeeld de gebruiksomvang tussen 90 en 110 procent van het vorige jaar). De risicofactor met betrekking tot de bandbreedte wordt verwerkt in de tarieven;
- het verbruik moet door een adequate registratie worden bewaakt, zodat een eventuele overschrijding van de bandbreedte tijdig wordt gesignaleerd;
- de 'underpinning contracts' met BI moeten worden afgestemd op de service level agreements die met de opdrachtgevers/afnemers worden afgesloten. Voor PB is het van belang dat minder opbrengsten gepaard gaan met minder kosten. Een beperkte uitbreiding van de werklast mag niet gepaard gaan met hoge extra kosten.

Tariefstelsel

De ingebruikname van een nieuw tariefstelsel is een risicofactor. Daarom is het noodzakelijk om de tariferingscyclus in het eerste jaar meerdere keren te doorlopen om eventuele ontsporingen tijdig te kunnen onderkennen en bij te stellen.

6.5 Ontwerp en implementatie tariefomgeving

De tariefomgeving is nodig om de doorbelasting goed te laten functioneren. Het gaat enerzijds om de benodigde technische voorzieningen en anderzijds om de administratieve organisatie. Technische voorzieningen bestaan onder andere uit:
- programmatuur waarmee de noodzakelijke doorbelastingsgegevens kunnen worden verzameld (TCS);
- een rekenmodel om de tariferingsregels op te stellen en te verifiëren (zie 6.4);
- aansluiting op een adequaat factureersysteem.

De administratieve organisatie behelst onder meer:
- beschrijven van de relevante procedures (inclusief taken en verantwoordelijkheden);
- opstellen werkinstructies:
  - wie is verantwoordelijk voor het verzamelen van welke gegevens;
  - op welke wijze komt de factuur tot stand;
  - wie controleert en accordeert de facturen;
  - wie rapporteert aan wie, waarover en hoe vaak;
  - welke vormen van overleg zijn er;
- ontwerpen documenten.

6.6 Evaluatie tariefstelsel

In de loop der tijd veranderen markten, doelstellingen, producten, processen, maar ook marktprijzen.

Voor ASZ kan de markt substantieel veranderen doordat er samenwerkingsverbanden ontstaan, bijvoorbeeld met Achmea. Marktprijzen kunnen veranderen door de komst van nieuwe hardware, die per geleverde prestatie-eenheid goedkoper is. Ook kunnen nieuwe producten ontstaan. Elk tariefstelsel moet daarom op gezette tijden worden geëvalueerd en zonodig worden herzien. Men moet zich dan onder andere afvragen:
- Voldoet het tariefstelsel nog aan de doelstelling, randvoorwaarden en omgevingsfactoren?
- Sluiten de doelstelling, randvoorwaarden en omgevingsfactoren nog optimaal aan bij de organisatie?
- Is er behoefte aan nieuwe diensten?
- Levert het tariefstelsel ongewenste effecten op?
- Zijn de intern doorbelaste kosten (door BI) nog marktconform?
- Sluit het tariefstelsel nog aan op de in rekening gebrachte kosten van externe leveranciers?

Vooral in het eerste jaar is het noodzakelijk om de tariferingscyclus meerdere keren te doorlopen om eventuele ontsporingen tijdig te kunnen onderkennen en bijstellen. PB zal binnenkort het nieuwe tariefstelsel evalueren. Het is echter ongewenst dat structuurveranderingen en tariefaanpassingen elkaar in een hoog tempo opvolgen. Daarom is het aan te bevelen om de tariferingscyclus op den duur niet vaker dan eens per jaar uit te voeren. Het inschakelen van een externe EDP-auditor kan daarbij zorgen voor een onafhankelijke en deskundige bijdrage.

7 Conclusie

In hoofdstuk 6 is een aantal omgevingsfactoren uit het ASZ businessplan 'Visie en strategie 1996-2000' genoemd, zoals:
- grotere klanttevredenheid bij het GAK;
- scherper inzicht in en beheersing van kosten;
- verdere verhoging en waarborging van de kwaliteit van de dagelijkse exploitatie;
- ASZ draagt bij aan effectiviteit, gebruikerstevredenheid en kostenbebeersing bij de gebruikersorganisatie;
- ASZ wil in het jaar 2000 een lager kostenniveau bereiken bij de inzet van automatiseringsmiddelen.

In hoofdstuk 2 zijn knelpunten en aanvullende eisen genoemd. In de hoofdstukken 5 en 6 is de oplossingsrichting geschetst: doorbelasten van het gebruik en beheer van de basisregistraties realiseren door de doorbelasting te richten op de diensten die worden verleend op de gegevensverzamelingen BRP, BRWG en BRDV en doorbelasten in voor gebruikers begrijpelijke eenheden, met name functionele transacties.

Het voordeel van de gekozen oplossing is dat door een adequate realisatie en implementatie een substantiële bijdrage wordt geleverd aan de verwezenlijking van het ASZ-businessplan 'Visie en strategie 1996-2000'; de gesignaleerde knelpunten zijn weggenomen en de aanvullende eisen zijn gerealiseerd.

8 Evaluatie

Het beschreven onderzoek is uitgevoerd in een aantal projectmatige stappen (zie tabel 3):

- in de periode juni/juli 1997 is door een externe IT-adviseur/IT-auditor een uitvoerige situatie- en knelpuntanalyse uitgevoerd. Daartoe is bestaande documentatie en literatuur bestudeerd en zijn er gesprekken gevoerd met zowel medewerkers van verschillende afdelingen van ASZ, als met medewerkers van enkele afdelingen van het GAK. De bevindingen zijn vastgelegd in een rapport. Bovendien zijn in het rapport oplossingsmogelijkheden geschetst en is een globale uitwerking gegeven van de oplossingsrichting;
- in de maand september 1997 is door de externe IT-adviseur/IT-auditor de oplossingsrichting verder uitgewerkt en is er een rekenmodel ter vaststelling van de tarieven ontwikkeld;
- in oktober/november 1997 heeft PB op basis van de aanbevelingen in bovenstaande rapporten enkele alternatieve invullingen van het aangedragen rekenmodel uitgewerkt. In overleg met het GAK is nog een aantal verfijningen van het model doorgevoerd en zijn de gewichtsfactoren bepaald;
- in december 1997 zijn de tarieven voor 1998 met behulp van het rekenmodel vastgesteld.

Hoe is het in het eerste kwartaal van 1998 gegaan met de nieuwe wijze van doorbelasten? Op deze vraag is in april 1998 antwoord gegeven door de heer Paul Peltzer, hoofd van het Productteam Basisregistraties. Het antwoord kan als volgt worden samengevat:

- enkele gebruikers moesten wennen aan het feit dat ze geen inzicht meer hebben in verbruikte prestatie-eenheden;
- de nieuwe wijze van doorbelasten heeft tussen applicaties, met betrekking tot de omvang van de doorbelastingskosten, grote verschuivingen gegeven. Het doorbelasten van het verbruik van functionele transacties leidt tot een duidelijkere berekening van de kosten;
- de afdeling GAK-IT, de IT-beleidsbepalende intermediair tussen het GAK en ASZ, is zeer enthousiast over de nieuwe wijze van doorbelasten omdat:
  - GAK-IT een ingang heeft gekregen om een beter inzicht te krijgen in het gebruik van de IT-voorzieningen binnen de GAK-organisatie;
  - GAK-IT de nieuwe doorbelastingsgegevens als basis kan gebruiken om met ASZ te overleggen over de wijze waarop het kostenniveau kan worden verlaagd (dit is een omgevingsfactor uit het ASZ businessplan);
  - discussies over zeer omvangrijke verschillen betreffende facturen tot het verleden behoren;
- binnen ASZ zijn de discussies met de toeleverende afdelingen van PB toegenomen omdat ook PB meer

Tabel 3 Projectmatige stappen van het onderzoek

Projectmatige stap	**Periode**	**Uitvoerder**
situatie- en knelpuntanalyse	juni/juli 1997	externe adviseur
globale uitwerking oplossingsrichting	juli 1997	externe adviseur
ontwikkelen rekenmodel ter vaststelling tarieven	september 1997	externe adviseur
uitwerken alternatieve invullingen rekenmodel	oktober/november 1997	ASZ in overleg met GAK
vaststellen tarieven voor 1998	december 1997	ASZ
evaluatie eerste kwartaal 1998	april 1998	ASZ en GAK
schrijven artikel	november 1997, april 1998	externe adviseur

inzicht heeft gekregen in de doorbelastingscomponenten;
- ASZ kan haar klanten een grotere flexibiliteit bieden omdat ASZ (in de betekenis van kosten) elk gewenst (deel)product kan leveren.

De discussies over de relaties tussen de dienstenprofielen en de producten zijn gestart. Paul Peltzer: De afgelopen jaren hebben we doorbelast op een manier die voor een belangrijk deel was gebaseerd op drijfzand. Met het 'rapport Van Dijk' hebben we nu een fundament onder ons beleid gelegd. Het beleid heeft 'smoel' gekregen. Het BSW-model en het drielagenmodel helpen ons om in de praktijk van basiskostendragers naar dienstenprofielen en eindgebruikersproducten te komen. Het onderzoek heeft dus niet alleen onze doorbelastingsproblematiek opgelost maar heeft ook een duidelijke spin-off opgeleverd.

Literatuurlijst

ASZ, 'ASZ wegwijzer', *ASZ*, mei 1997.

Bladergroen, D. et al, 'Planning en beheersing van IT-dienstverlening', *Kluwer BedrijfsInformatie*, Deventer, 1995.

Biene-Hershey, M.E. van en M.A.Bongers, 'Managers doen steeds vaker beroep op EDP-auditors', *Automatisering Gids*, 6 juni 1997, pag. 15.

Bras, G., 'Doorbelasten van kosten blijft steken in woorden', *Automatisering Gids*, 4 juli 1997, pag. 17/23.

Dijk, A.J. van en P. Peltzer, 'Doorbelasting van kosten van gebruik en beheer (exploitatie en onderhoud) van de basisregistraties aan afnemers, Situatie- en knelpuntanalyse, oplossingsrichting', *ASZ/PB*, september 1997.

Dijk, A.J. van, P. Kooij en P. Peltzer, 'Doorbelasting van kosten van gebruik en beheer (exploitatie en onderhoud) van de basisregistraties aan afnemers, Vaststellen tarieven, stappenplan/rekenmodel', *ASZ/PB*, september 1997.

Directie ASZ, 'Businessplan ASZ, Visie en strategie 1996-2000', *ASZ*, Amsterdam, november 1996.

Evelo, M., M. van den Ham en E.J. Swets, 'Doorbelasting van IT-kosten', *Informatie*, december 1995, pag. 804-809.

Gleusteen, S.E. en M.H.C. Huygen, 'Doorbelasting Automatiseringskosten GAK (DAG), Onderzoek naar de kostenregistratie en doorbelasting van computerkosten', *GAK, AUT/Research & Development*, mei 1991.

Janszen, G.A.B. et al, 'Doorbelasting van kosten van geautomatiseerde informatievoorziening', *Kluwer Bedrijfswetenschappen*, Deventer, 1987.

Johnson, B., 'Cost Management for IT Services, IT Infrastructure Library', *CCTA Library*, Norwich, 1991.

Kind, R. en M. van Leeuwen, 'Doorbelasten van kosten is onderschat beleidsinstrument', *Automatisering Gids*, 18 juli 1997, pag. 13.

Looijen, M., 'Beheer van Informatiesystemen', *Kluwer Bedrijfsinformatiesystemen*, Deventer, 1997.

Mersel, R.J. en J. de Boer, 'Gedistribueerde databases in de praktijk: de BasisRegistratie Personen bij het GAK', *IT Management*, 1995/1, pag. 59-67.

Niekel, P. en M. van Wonderen, 'Service Management Informatiesystemen, Helpdesk bij GAK', *IT Beheer Praktijkjournaal*, juni 1997, pag. 5-7.

auteur

Ir. Aart J. van Dijk RE RI is werkzaam als onafhankelijk IT-adviseur/IT-auditor bij Avédé-Info b.v. te Zoetermeer.

Publication 7

IMPALA

Een beheersingsmodel voor de

bedrijfstelefonie-services

Management Control System for company Telephony Services

1994

The expression:
"the client is not waiting for an assessment but especially for advice about how the situation can be improved" is also applicable outside the domoin of IT auditing.

Aart van Dijk

This article previously appeared in:

Informatie, December 1994, pages 843- 855

Abstract

In 1993, the Delft University of Technology (DUT) carried out the project 'Vervanging Telefooncentrale' (Replacement Telephone Exchange). As became clear from literature studies and conversations with advisors and suppliers in the field of private branch exchanges, not a lot was organised with regard to the management of PABXs (voice communication). Besides, the role of the private branch exchange is changing. In this article, this changing role is examined and the IMPALA control model is described. In this model, the theory of the Rotterdam School (EDP auditing Erasmus University Rotterdam) plays an important part. IMPALA is presented as a framework and may act as a basis for controlling the private branch exchange and the Telephony Services that are part of this.

IMPALA

Een beheersingsmodel voor de bedrijfstelefonie-services

Aart J. van Dijk

De Technische Universiteit Delft heeft in 1993 het project 'Vervanging Telefooncentrale' uitgevoerd. Op het terrein van het beheer van PABX'en (spraakcommunicatie), zo bleek uit literatuurstudie en gesprekken met adviseurs en leveranciers op het gebied van bedrijfstelefooncentrales, is nog weinig geregeld. Bovendien verandert de rol van de bedrijfstelefooncentrale. In dit artikel wordt op deze veranderende rol ingegaan en wordt het beheersingsmodel IMPALA beschreven. Bij dit model speelt de theorie van de Rotterdamse school (EDP-auditing Erasmus Universiteit Rotterdam) een belangrijke rol. IMPALA wordt als raamwerk gepresenteerd en kan als basis fungeren voor de beheersing van de bedrijfstelefooncentrale en de daarbij behorende Telefonie Services.

1 Inleiding

De Technische Universiteit Delft (TU Delft) heeft in 1993 het project 'Vervanging Telefooncentrale (VTC)' uitgevoerd. Het doel van het project was het in produktie brengen van een nieuwe telefooncentrale (PABX) met randapparatuur, die voldoet aan de functionele eisen van de komende 7 à 8 jaar en het aanbieden van de Telefonie Services (TS) als dienst van het Rekencentrum.

Ten behoeve van de fase 'invoering' was een projectorganisatie geformeerd, die naast de stuurgroep en de projectcoördinatie bestond uit de volgende werkgroepen:

- Marketing, Voorlichting & Opleiding;
- Inventarisatie & Implementatie;
- Functies;
- Beheer & Bediening.

De taak van de werkgroep Beheer & Bediening was de ontwikkeling van procedures en gereedschappen ten behoeve van beheer en bediening, het opstellen van afspraken over de wijze waarop het beheer van het PABX-systeem georganiseerd zou worden en de voorbereiding van de overdracht van deze zaken aan de staande organisatie.

Uit literatuurstudie en gesprekken met adviseurs en leveranciers op het gebied van bedrijfstelefooncentrales bleek dat met betrekking tot het beheer van PABX'en (spraakcommunicatie) nog weinig is geregeld. Bovendien verandert de rol van de bedrijfstelefooncentrale.

Dit artikel heeft als doel inzicht te geven in de veranderende rol van de bedrijfstelefooncentrale (PABX) en aan te geven hoe de PABX en de daarbij behorende Telefonie Services kunnen worden beheer(d)(st). Daarbij wordt de theorie van de *Rotterdamse school* (daaronder wordt verstaan de theorievorming rondom en over EDP-auditing, zoals deze wordt gedoceerd aan de Erasmus Universiteit Rotterdam) 'losgelaten' op de bedrijfstelefonie-services. De invalshoek is de beheersing vanuit de *gebruikersorganisatie (GO)*.

Paragraaf 2 beschrijft de historische rol van de (bedrijfs)telefooncentrale, terwijl in paragraaf 3 wordt ingegaan op recente ontwikkelingen. De beheersingsproblematiek komt in paragraaf 4 aan de orde. Objecten, beheersmaatregelen en kwaliteitsaspecten worden beschreven. Tezamen met een Management Control System wordt hiermee het beheersingsmodel IMPALA geconstrueerd. Dit als raamwerk gepresenteerde model moet in de praktijk worden uitgewerkt.

In paragraaf 5 wordt het model IMPALA op enkele punten uitgewerkt voor de TU Delft. Een korte samenvatting is opgenomen in paragraaf 6.

2 Historische rol van de (bedrijfs)telefooncentrale

De introductie van de telefoon

Al is het discutabel wie de telefoon (spreken op afstand) heeft uitgevonden (Reis, Gray en Bell zijn de meest serieuze kandidaten), het was Alexander Graham Bell (1847-1922), zoon en kleinzoon van spraakleraren, die op 7 maart 1876 patent verkreeg op een principe dat het mogelijk maakte om op eenvoudige wijze de menselijke stem een grotere afstand te laten overbruggen dan door roepen mogelijk was. In nog zeer onvolmaakte vorm verscheen het hiervoor vereiste apparaat in het voorjaar van 1876 op de Centennial Exhibition te Philadelphia (Hogesteeger, 1976).

Vanuit de Verenigde Staten bereikte de telefoon via Engeland en Duitsland ons land. Het eerste openbare telefoonnet kwam in juni 1881 in Amsterdam tot stand en telde 48 abonnees. De woningen van de eerste abonnees waren via bovengrondse draden verbonden met de telefooncentrale. Het net behoorde aan de Nederlandsche Bell Telefoon Maatschappij, die van de gemeente een concessie verkregen had. Na veel juridisch touwtrekken verkregen ook andere steden hun telefoonnetten. De eerste telefooncentrales waren handcentrales (Stuvel, 1952), naderhand kwamen de geautomatiseerde en computergestuurde centrales.

De bedrijfstelefooncentrale

Telefoongesprekken werden niet alleen door privé-personen gevoerd, maar ook bedrijven maakten steeds meer gebruik van de telefoon. Voor bedrijven kan het financieel interessant zijn om een eigen bedrijfstelefooncentrale te bezitten. Met behulp van een PABX, Private Automatic Branch eXchange, ofwel bedrijfstelefooncentrale, kan een bedrijf namelijk het aantal verbindingen (netlijnen) met de openbare communicatie-infrastructuur beperken. Zo kan een bedrijf met honderden telefoontoestellen volstaan met enkele tientallen netlijnen. Het aantal netlijnen bepaalt het aantal externe gesprekken dat tegelijkertijd gevoerd kan worden. Een tweede financieel voordeel is dat interne gesprekken geen gesprekskosten met zich meebrengen, omdat deze gesprekken geen gebruik maken van de openbare communicatie-infrastructuur.

3 Recente ontwikkelingen

De groei van zowel spraak- als 'niet-spraak'-communicatie is mede het gevolg van ontwikkelingen op het gebied van de datacommunicatie, de facsimile, de kantoorautomatisering met personal computers met datacommunicatiefaciliteiten en de value added services die in toenemende mate in het telecommunicatienetwerk zijn opgenomen. Op een aantal nieuwe ontwikkelingen wordt hier kort ingegaan.

Facsimile

In de telecommunicatie wordt onder facsimile (FAX) verstaan: het via een telecommunicatieverbinding overseinen en het aan de overzijde van de verbinding vervaardigen van een kopie van een origineel. De belangrijkste soort van facsimile is de momenteel erg populaire documentfacsimile (Vonk-Wiersema, 1990). Het telefoonnet is het draagsysteem voor facsimile.

Paging-systeem

Bereikbaarheid van ambulante personen binnen een bedrijf is vaak een probleem. Voor dergelijke gevallen kan aan de PABX een personenoproepsysteem (ook wel 'paging' genoemd) worden gekoppeld (Van Heck, 1989). Vanaf ieder willekeurig toestel kunnen dan ambulante personen, die een pager ('pieper') bij zich dragen, worden opgeroepen. Daartoe draait men het nummer van het paging-systeem, gevolgd door de persoonlijke oproepcode en het toestelnummer waarmee de opgeroepene contact moet opnemen. Het paging-systeem 'piept' nu de betrokkene op. Deze kan vervolgens op zijn pager zien welk toestelnummer hij moet bellen.

Automatic Call Distribution

Binnen bepaalde afdelingen in een bedrijf kunnen intensieve verkeersstromen ontstaan met de buitenwereld. Deze intensieve verkeersstromen vragen om bijzondere communicatievoorzieningen, waarbij de oproepen over de aanwezige medewerkers worden verdeeld (Van Wijnen, 1993). Dergelijke voorzieningen worden ook wel ACD (Automatic Call Distribution) genoemd. Op afdelingen die intensief extern worden geraadpleegd, zoals de afdeling Studentenadministratie van de TU Delft, zijn geïntegreerde ACD-voorzieningen ingezet.

Voice Mail

Antwoordapparaten komen in de particuliere sector steeds meer voor. De beller kan met behulp van een antwoordapparaat bij afwezigheid van de opgebelde een gesproken bericht achterlaten.
Door een zogenaamde voice mail-eenheid te koppelen met de bedrijfstelefooncentrale kan op soortgelijke wijze een boodschap worden ingesproken. De oproeper wordt bij afwezigheid van een toestelgebruiker doorgeschakeld naar een persoonlijke postbus in de voice mail-eenheid en kan dan een boodschap inspreken. Met behulp van message waiting-signalen op de toestellen worden de eigenaren van de postbussen op de hoogte gesteld van de aanwezigheid van berichten.

ISDN

Sedert 1984 worden er in Japan, de Verenigde Staten en Europa proeven gedaan met een nieuw 'telefoonnet'. Dit nieuwe net is thans in een vergevorderd stadium en nadert zijn voltooiing in de moderne landen.
Het net zal niet alleen telefonie (spraak), maar ook data en beelden transporteren. Verschillende informatiediensten, die nu onafhankelijk van elkaar plaatsvinden, worden dan geïntegreerd in één net. Dit nieuwe net wordt ISDN (Integrated Services Digital Network) genoemd. In ISDN is voorzien dat iedere abonnee ten minste een aderpaar krijgt met twee basiskanalen van 64 kbit/s plus een kanaal van 16 kbit/s ten behoeve van signaleringsinformatie. ISDN zal in een aantal opzichten verschillen van de huidige situatie:

- de transmissiesnelheid wordt veel groter;
- diensten worden toegankelijk via één net;
- er worden nieuwe diensten mogelijk.

Eén van die nieuwe mogelijkheden is calling line-identificatie (cli) waardoor de telefoonabonnee kan zien vanaf welk nummer hij wordt gebeld (Verstraaten, 1993). Het aantal mogelijkheden met ISDN is praktisch onbegrensd (spraak, data, beeld, geïntegreerde toepassingen). Met name het vakgebied informatica is van cruciaal belang voor de opkomst van ISDN (Bruijning, 1992). Daarbij gaat het vooral om het ontsluiten van ISDN door het aanbieden van nieuwe toepassingen, zoals telewerken, toegang tot (grafische) databanken en beeldgeoriënteerde toepassingen.

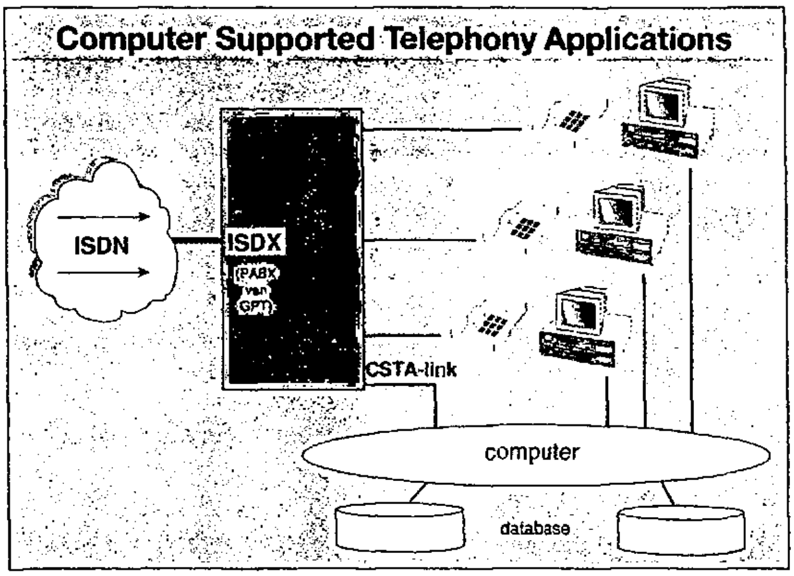
Computer Integrated Telephony

Steeds meer organisaties gaan ertoe over de contacten met hun relaties via de telefoon af te handelen zonder tussenkomst van een medewerker en gebruiken daarvoor een voice response-systeem. Bij dergelijke toepassingen wordt het telefoontoestel gebruikt als terminal. De sprekende computer gebruikt kleine, op schijf opgeslagen teksten, die in een studio zijn ingesproken. Deze vaste elementen worden gebruikt om woorden uit te spreken

en kleine standaardzinnen te formuleren. De spraak wordt gegenereerd en informatie wordt verstrekt aan de hand van de acties die de vragensteller met zijn telefoontoestel onderneemt (Sunter, 1991).
Een voice response-systeem is één van de vormen van samenwerking tussen spraak (telefoonsysteem) en data (computersysteem). Er is sprake van integratie in technische en functionele zin.
Namen die daarvoor worden gebruikt, zijn onder andere 'Computer Integrated Telephony (CIT)' en 'Computer Supported Telephony Applications (CSTA)' (Walters, 1993).
De organisatie krijgt hiermee de mogelijkheid om telefoonfuncties uit te voeren in samenwerking met de faciliteiten van een computersysteem, zoals krachtige en flexibele databases. Figuur 1 geeft aan hoe de relatie tussen telefonie en computer is geregeld (bron: GPT – General Plessey Telecommunications).

De toekomst

Telecommunicatie staat momenteel sterk in de belangstelling, vooral in combinatie met het vakgebied informatica.
De combinatie, het vakgebied *telematica*, komt bijna wekelijks aan de orde in dagbladen, weekbladen en tv-programma's. De term 'Electronic Highway' wordt daarbij vaak gebruikt. Een bespreking hiervan valt buiten het kader van dit artikel.
Voor bedrijven zijn twee onderwerpen van bijzonder belang, te weten: mobiele telefonie (Leenhouts, 1994) en multimedia. Voorbeelden van multimedia-toepassingen zijn beeldtelefoon, videovergaderen en tele-educatie (leren op afstand).



Figuur 1: Computer Integrated Telephony van GPT

Bevindingen naar aanleiding van recente ontwikkelingen

In deze paragraaf is de veranderende rol van de bedrijfstelefooncentrale (PABX) beschreven. Van een handgeschakelde telefooncentrale is de PABX veranderd in een software-bestuurde centrale, die een essentieel onderdeel uitmaakt van de, soms geïntegreerde, data- en telecommunicatievoorzieningen van een bedrijf. Het is daarom merkwaardig dat nogal wat bedrijven over een computercentrum beschikken dat aan alle daaraan te stellen eisen voldoet, maar tegelijkertijd de bedrijfstelefooncentrale ergens in een stoffige, niet beveiligde kelder hebben staan. Deze bedrijven zijn echter vaak sterk afhankelijk van hun telefonie-services.

Een kenmerkende invloed van automatisering is het ontstaan van twee additionele functionele organisaties naast de gebruikersorganisatie (GO). Deze organisaties zijn de organisatie voor systeemontwikkeling en systeemonderhoud (SO) en de verwerkings- en transporterende organisatie (VTO). Binnen de GO worden bovendien gegevensbeheer (GB) en applicatiebeheer (AB) als verbijzonderde functies onderkend. SO, VTO, GO-GB en GO-AB vormen tezamen de Ondersteunende Organisatie van de Informatie Voorziening (OOIV).
Ten behoeve van de telefonie-services moet een OOIV worden ingericht. Bovendien moeten de PABX en de daarbij behorende telefonie-services worden beheerst op de volgende aspecten (EUR, 1993):
- effectiviteit/doeltreffendheid;
- betrouwbaarheid;
- continuïteit;
- vertrouwelijkheid;
- efficiency/doelmatigheid.

Het hiervoor noodzakelijke beheersingsmodel wordt hierna uitgewerkt.

4 Een beheersingsmodel voor de Telefonie-Services

Het formuleren van wat onder beheer wordt verstaan, is niet eenvoudig vanwege de vele omschrijvingen die worden gehanteerd. Een praktische formulering is te vinden bij Looijen & Delen (1992): 'het in een zodanige toestand houden van een voorziening dat deze blijft voldoen aan de vastgelegde eisen en de vanzelfsprekende behoeften van de gebruikers.' Hierbij moet zowel worden gedacht aan de functionele eisen als aan de kwaliteitseisen. *Beheer* van telefonie-services is niet alleen besturing van en toezicht houden op, maar ook het *beheersen* – in de zin van meester zijn – van technische, organisatorische, economische en personele aspecten.

In de vorige paragrafen is aangegeven dat de rol van de bedrijfstelefooncentrale is veranderd en is de relatie gelegd met de informatietechnologie. Bedrijven zijn vaak sterk afhankelijk van hun telefonie-services (TS), doch dit kan per bedrijf verschillen. Als eerste stap dient de *mate van afhankelijkheid* van de TS te worden vastgesteld. Een verzekeringsbedrijf met als belangrijk primair proces 'telefonische verkoop' is bijvoorbeeld meer afhankelijk van de TS dan de gemiddelde onderwijsinstelling. In deze paragraaf wordt aangegeven hoe met behulp van het KAD-model (kwaliteit van administratieve dienstverlening; zie Hartog, Molenkamp & Otten, 1992) de keuze van de objecten die behoren tot de TS, tot stand kan komen en worden de door de TU Delft bepaalde *objecten* opgesomd. Bovendien wordt ingegaan op de *beheersmaatregelen* en wordt de verbinding gemaakt met het *Management Control System* (Kocks, 1992*a*). Ook wordt in deze paragraaf ingegaan op de relatie tussen de objecten en de kwaliteitsaspecten effectiviteit, betrouwbaarheid, continuïteit, vertrouwelijkheid en efficiency. Uitgangspunt daarbij is dat de problematiek wordt benaderd en beschouwd vanuit de GO. Het geheel wordt als een raamwerk weergegeven in een model dat de naam IMPALA (*I*nformation *M*anagement *PA*BX and *L*ocal *A*dministration) heeft gekregen en dat in de praktijk verder kan worden uitgewerkt.

Keuze van objecten

Produkten, waaronder inbegrepen diensten, vormen de bestaansgrond van een organisatie en zijn het resultaat van de strategische besluitvorming door het management over '*wat*' de organisatie produceert en welke de eisen zijn waaraan die produkten moeten voldoen. Nadat besloten is '*wat*' de organisatie wil realiseren, kunnen vervolgens de bijbehorende processen (het '*hoe*') worden bepaald. In het KAD-model wordt uitgegaan van het *systeembegrip*. Het systeem wordt gedefinieerd en afgebakend door de systeemgrens, welke bepalend is voor wat als invoer en wat als uitvoer wordt beschouwd. Het produkt vormt de uitvoer van het systeem en bepaalt zodoende een deel van de grens van het systeem waarbinnen dat proces wordt uitgevoerd.
Het *bepalen van produkten* kan als volgt plaatsvinden:

1 selecteren van de produkten die nader worden geanalyseerd;
2 bepalen van de informatiebehoeften;
3 analyseren en vaststellen van de produktnormen.

De selectie van produkten vindt plaats door afnemersgroepen, produkten en kritische produkten vast te stellen. Bij het vaststellen van produkten is het van belang de produkten aan de afnemersgroepen te relateren (Hartog, Molenkamp & Otten, 1992). Per produkt worden de (interne en externe) afnemers aangegeven. Daarbij is het van belang rekening te houden met de functie van het produkt voor de gebruikers. Waarvoor gebruiken zij het produkt; waarom hebben zij het produkt nodig? Het is mogelijk dat één produkt diverse functies vervult voor verschillende afnemersgroepen.
Het bepalen van de informatiebehoeften heeft tot doel de kwaliteit van het produkt zichtbaar te maken en vormt de basis voor het vaststellen van de produktnormen. Het is de taak en verantwoordelijkheid van het management de kritische succesfactoren te bepalen en deze te (laten) vertalen naar operationele produktnormen.

Voor het *bepalen van processen* zijn eveneens drie stappen te onderscheiden:

1 selecteren en afbakenen van de processen;
2 analyseren van de processen;
3 vaststellen van de inrichting van de processen.

Bij het selecteren van de processen spelen de kritische processen van de organisatie een belangrijke rol. Het analyseren van het proces betekent dat wordt ingezoomd op het proces. Daarbij wordt het proces uitvergroot tot op 'deelprocesniveau'. Bij het analyseren worden de beheersmaatregelen in kaart gebracht.

Nadat alle processen zijn geanalyseerd en ingericht, is er sprake van twee soorten beheersmaatregelen:

1 algemene beheersmaatregelen;
2 toepassingsgerichte beheersmaatregelen per object.

Naast produkten en processen spelen binnen een organisatie de structuur en de middelen (het '*waarmee*') een belangrijke rol (Kocks, 1991). Met *structuur* wordt bedoeld de wijze waarop binnen een organisatie bepaalde taken, met bijbehorende verantwoordelijkheden en bevoegdheden, zijn samengevoegd tot functies die op zich weer zijn ondergebracht in afdelingen. Binnen deze afdelingen werken personen aan wie die taken weer zijn toebedeeld. Het samenstel van procedures completeert het geheel.
Middelen zijn de componenten die deel uitmaken van een proces. Voorbeelden van middelen zijn: grondstoffen, machines, mensen en informatie.
Objecten kunnen bestaan uit structuur, middelen, processen, produkten of een mix van deze vier categorieën.

Bij de TU Delft zijn ten behoeve van de TS de volgende objecten bepaald:

1 gesprekken;
2 faciliteiten;
3 telefoongids;
4 onderhoud;

5 problem management (storingen, enzovoort);
6 bekabeling;
7 change management (opdrachten/mutaties);
8 ACD;
9 voice mail/response;
10 Service Level Agreement;
11 doorberekening;
12 alarmeringsprocedures;
13 bediening;
14 paging;
15 CIT-applicaties;
16 inkoop/verkoop toestellen;
17 overdrachtprocedures;
18 calamiteitenplan;
19 beheersorganisatie;
20 Handboek Telefonie Services.

Toelichting (zie ook tabel 1)

1 Binnenkomende gesprekken (object 1)
Een relatie van de TU Delft kan een personeelslid rechtstreeks bellen als hij het betreffende doorkiesnummer kent. Kent hij dit nummer niet, dan kan hij zijn doel bereiken door de telefoongids te raadplegen (object 3) of via het centrale aankiesnummer en de telefoniste (object 13).
Indien het betreffende personeelslid afwezig is, dan kan de relatie worden doorgeschakeld naar het 'toestel bij afwezigheid' van het personeelslid (object 2) of worden verbonden met de voice mail-eenheid (object 9) om een boodschap in te spreken.
Voor sommige afdelingen komen, in bepaalde perioden, zoveel telefoongesprekken binnen dat het noodzakelijk is gebruik te maken van bijzondere communicatievoorzieningen, zodat de gesprekken worden verdeeld over de medewerkers (object 8). In sommige gevallen is het gewenst 'klantgegevens' van de beller onmiddellijk beschikbaar te hebben. Dit kan worden geregeld met behulp van calling line-identificatie en Computer Integrated Telephony (object 15).

	structuur	middel	proces	produkt
GO	19	3,12,20	2,7,8, 9,13,14	1
SO			17	
VTO	19	6,10, 15,18	4,5,11, 16,17	
	(waarmee)	(waarmee)	(hoe)	(wat)

Tabel 1: Objecten Telefonie Services TU Delft

2 Uitgaande gesprekken (object 1)
Een personeelslid die een (buitenlandse) relatie wil bellen, kan dat rechtstreeks doen als zijn verkeersklasse dat toestaat (object 2) of kan via de telefoniste het gesprek tot stand laten brengen (object 13). Dit laatste is bij de TU Delft slechts toegestaan als de betreffende beheerseenheid (faculteit/dienst) afspraken daartoe heeft gemaakt met de sectie 'Telefonie Services' (object 19, object 10). De kosten van de gesprekken, alsmede de bemiddelingskosten, worden aan de beheerseenheden doorberekend (object 11). Sommige relaties of collega's hebben een ambulante functie en zijn bereikbaar via een paging-systeem (object 14). Met name collega's die worden ingeschakeld in het geval van een brandmelding of een oproep om EHBO te verlenen (object 12) beschikken over een 'pieper'.

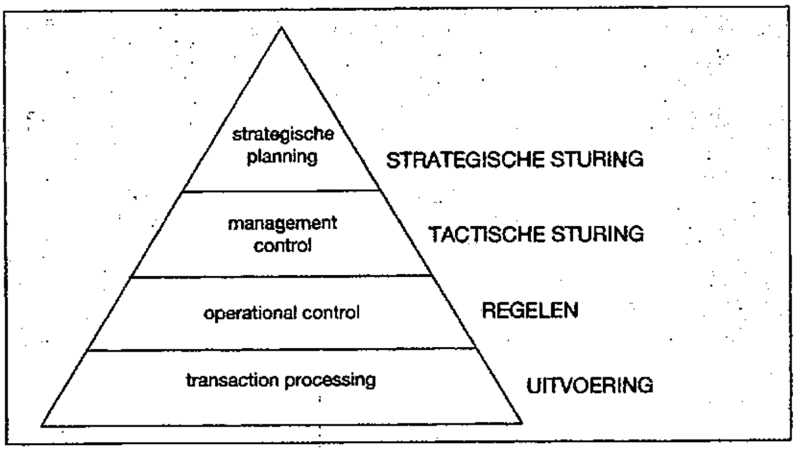
3 Diversen
Nieuwe medewerkers moeten worden voorzien van een telefoontoestel (object 16). Verhuizende medewerkers willen vaak op hun nieuwe werkplek hun 'eigen' nummer behouden. Verzoeken worden door middel van opdrachten tot mutaties (object 7) ingediend. Sommige verzoeken leiden tot aanpassing van de bekabeling (object 6).
Met de beheerseenheden zijn afspraken gemaakt over de beschikbaarheid van de diverse componenten van de Telefonie Services (object 10). Daartoe is het onder andere noodzakelijk dat preventief onderhoud (object 4) wordt uitgevoerd en dat procedures voor problem management (object 5) zijn geïmplementeerd.
Zowel met betrekking tot systeemsoftware (PABX) als applicatiesoftware (zoals doorberekening en gids) zijn overdrachtprocedures opgesteld (object 17). Om calamiteiten het hoofd te kunnen bieden moet men een calamiteitenplan beschikbaar hebben (object 18). Alle belangrijke zaken betreffende de Telefonie Services moeten worden vastgelegd in een Handboek Telefonie Services (object 20).

Beheersmaatregelen

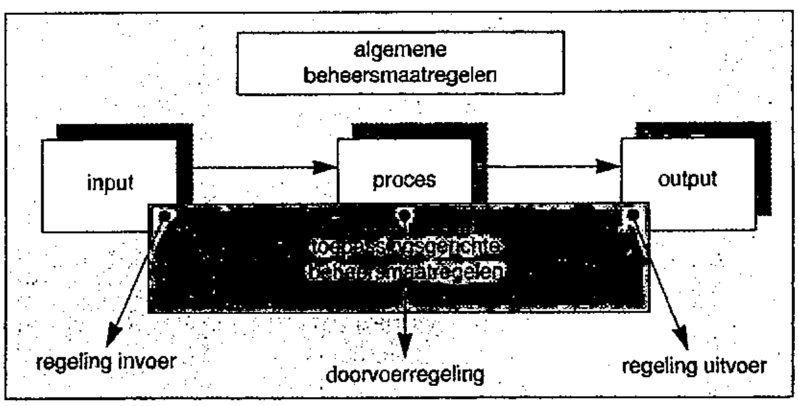
Sturen en regelen vormen de besturings- en beheersmechanismen van de uitvoering van het proces, elk met een eigen functie. *Sturen* is gericht op een blijvend voorzien in de veranderende behoeften van de omgeving, door middel van het vaststellen en zo nodig aanpassen van de functie en inrichting van het systeem, terwijl *regelen* gericht is op voortdurend realiseren van de systeemfunctie (Hartog, Molenkamp & Otten, 1992). De besturings- en beheersmechanismen zijn besluitvormingsprocessen. Zij worden gekenmerkt door het nemen van beslissingen, waarbij gebruik wordt gemaakt van informatie. Er wordt onderscheid gemaakt tussen *operationele informatie*, die

wordt gebruikt bij de uitvoering van een proces, en *managementinformatie*. Met managementinformatie wordt zowel gedoeld op de *verantwoordingsinformatie*, ten behoeve van de tactische en strategische sturing, als op de *beheersinformatie* ten behoeve van de verschillende regelfuncties. De besturings- en beheersmechanismen zijn zodoende bepalend voor de inrichting van de benodigde bestuurlijke informatievoorziening.
Davis en Olson (1985) geven verschillende soorten beslissingen weer in een beslissingshiërarchie (zie figuur 2).

Het regelen kan per geselecteerd proces plaatsvinden met behulp van de KAD-procesmodule. Door meten en reageren op fluctuaties in de invoer, de doorvoer en de uitvoer van het proces wordt ervoor gezorgd dat de output van het systeem blijft voldoen aan de door de stuurfunctie vastgestelde normen (zie figuur 3). Per proces moet managementinformatie (verantwoordingsinformatie) worden verzameld ten behoeve van de *tactische sturing* (informatie over de mate waarin de procesnormen worden gerealiseerd) en de *strategische sturing* (informatie over de mate waarin de produktnormen worden gerealiseerd) met behulp van het *Management Control System*.



Figuur 2: Beslissingshiërarchie (bron: Davis & Olson, 1985)



Figuur 3: Beheersmaatregelen

Beheersmaatregelen zijn maatregelen die het management moet nemen om een beheersbare organisatie met een goed functionerende informatievoorziening te verkrijgen. De beheersmaatregelen kunnen in drie groepen worden verdeeld (Van Praat & Suerink 1992):
- organisatorische beheersmaatregelen;
- procedurele beheersmaatregelen;
- technische beheersmaatregelen.

Organisatorische beheersmaatregelen
Binnen een organisatie wordt aan bepaalde functionarissen een bepaalde functie toegewezen met de daarbij behorende bevoegdheden en verantwoordelijkheden. Om alle functionarissen op een goede manier met elkaar te laten samenwerken, dient het management te zorgen voor een bepaalde organisatiestructuur. De taakverdeling binnen de organisatie moet zodanig ingericht zijn dat functies worden gecreëerd waarbij sprake is van *controletechnische functiescheidingen*. Het gaat om de volgende soorten functies: beschikkende functies, bewarende functies, registrerende functies, uitvoerende functies en controlerende functies (Jans, 1989).

Procedurele beheersmaatregelen
Met functiescheiding alleen kan niet worden volstaan. Er zijn ook procedurele beheersmaatregelen nodig om een adequate en verantwoorde functie-uitoefening te realiseren. Het gaat daarbij om het interne afstemmingsprobleem. In de eerste plaats moeten er procedures zijn om de uit de functie voortvloeiende bevoegdheden en verantwoordelijkheden te kunnen realiseren.
Dergelijke procedures worden vastgelegd in *functie- en taakomschrijvingen* die specifiek gericht zijn op een bepaalde functie, terwijl ook algemene richtlijnen bij de functie-uitoefening van belang kunnen zijn. Hierbij valt te denken aan zaken als werktijden, opnemen van vakantiedagen, enzovoort. Een tweede, zeker in het kader van de beheersing, belangrijk aspect, betreft de *coördinatie* tussen de functies.

Technische beheersmaatregelen
Met betrekking tot de technische beheersmaatregelen kan een onderscheid gemaakt worden tussen fysieke en logische beheersmaatregelen. Bij *fysieke beheersmaatregelen* wordt gebruik gemaakt van zichtbare en ook tastbare middelen. *Logische beheersmaatregelen* hebben veelal te maken met de software. Bekende voorbeelden zijn toegangsbeveiligingspakketten en encryptie-software.

Kwaliteitsaspecten versus objecten

In dit artikel worden de kwaliteitsaspecten die in de opleiding in EDP-auditing aan de EUR uitvoerig worden behandeld, als uitgangspunt genomen. Deze kwaliteitsaspecten zijn:

- effectiviteit;
- betrouwbaarheid;
- continuïteit;
- vertrouwelijkheid;
- efficiency.

aspecten	effectiviteit			betrouwbaarheid			continuïteit			vertrouwelijkheid			efficiency		
objecten	go	so	vto	go	so	vto	go	so	vto	go	so	vto	go	so	vto
gesprekken															
faciliteiten	×														
telefoongids															
onderhoud															
problem management															
bekabeling													×		
change management															
ACD															
voice mail/response	×														
SLA															
doorberekening				×						×					
alarmeringsprocedures															
bediening															
paging-systeem															
CIT-applicaties															
inkoop/verkoop toestellen															
overdrachtsprocedures															
calamiteitenplan							×								
beheersorganisatie															
handboek telefonie services															

× = in de tekst uitgewerkte object/aspect-combinatie

Figuur 4: Relatie aspect versus object binnen de Telefonie Services van de TU Delft

Er zijn geen algemene definities van kwaliteitsaspecten. Elke relatie aspect/object kent haar eigen omgeving en is situationeel bepaald.

Figuur 4 bevat de eerder genoemde twintig objecten die zijn uitgezet tegen de kwaliteitsaspecten. In dit artikel wordt de problematiek benaderd vanuit de GO. Dat neemt niet weg dat niet volledig voorbij kan worden gegaan aan de SO en de VTO. Daarom zijn de drie organisaties in figuur 4 opgenomen. Uit de figuur zou de conclusie getrokken kunnen worden dat ieder aspect aan ieder object kan worden gekoppeld. Dit is echter niet het geval. Zo is het niet zinvol om te praten over de 'vertrouwelijkheid van de beheersorganisatie' of de 'efficiency van het object Service Level Agreement'.

In paragraaf 5 wordt op enkele object/aspect-combinaties ingegaan. Een volledige uitwerking van alle relevante object/aspect-combinaties voor zowel de GO, SO als VTO valt buiten het bestek van dit artikel.

Management Control System

De beheersmaatregelen zijn in deze paragraaf aan de orde gekomen. Het betreft een middelen/maatregelenmix (zowel organisatorisch als fysiek, op het terrein van hardware en software) die is ontstaan bij het ontwikkelen van een organisatie.

Na het ontwikkelproces zal de middelen/maatregelenmix moeten worden onderhouden.

Per proces wordt managementinformatie (verantwoordingsinformatie) verzameld ten behoeve van de tactische en strategische sturing, die kan worden ingevuld door het *Management Control System*. Dit control system bestaat uit drie delen:

- *het monitoring system:* het stelsel van beheersmaatregelen om op grond van specifieke informatie vast te stellen dat processen, informatie en maatregelen voldoen aan de gestelde eisen;
- *het feedback system:* het volgens vaste regels informeren van het hogere echelon over de resultaten van de monitoring;
- *het response system:* de wijze waarop en de mate waarin het management al dan niet overgaat tot bijsturing (Kocks, 1992*a*).

In figuur 5 is het Management Control System voor de Telefonie Services (MCS-TS) van de TU Delft weergegeven.

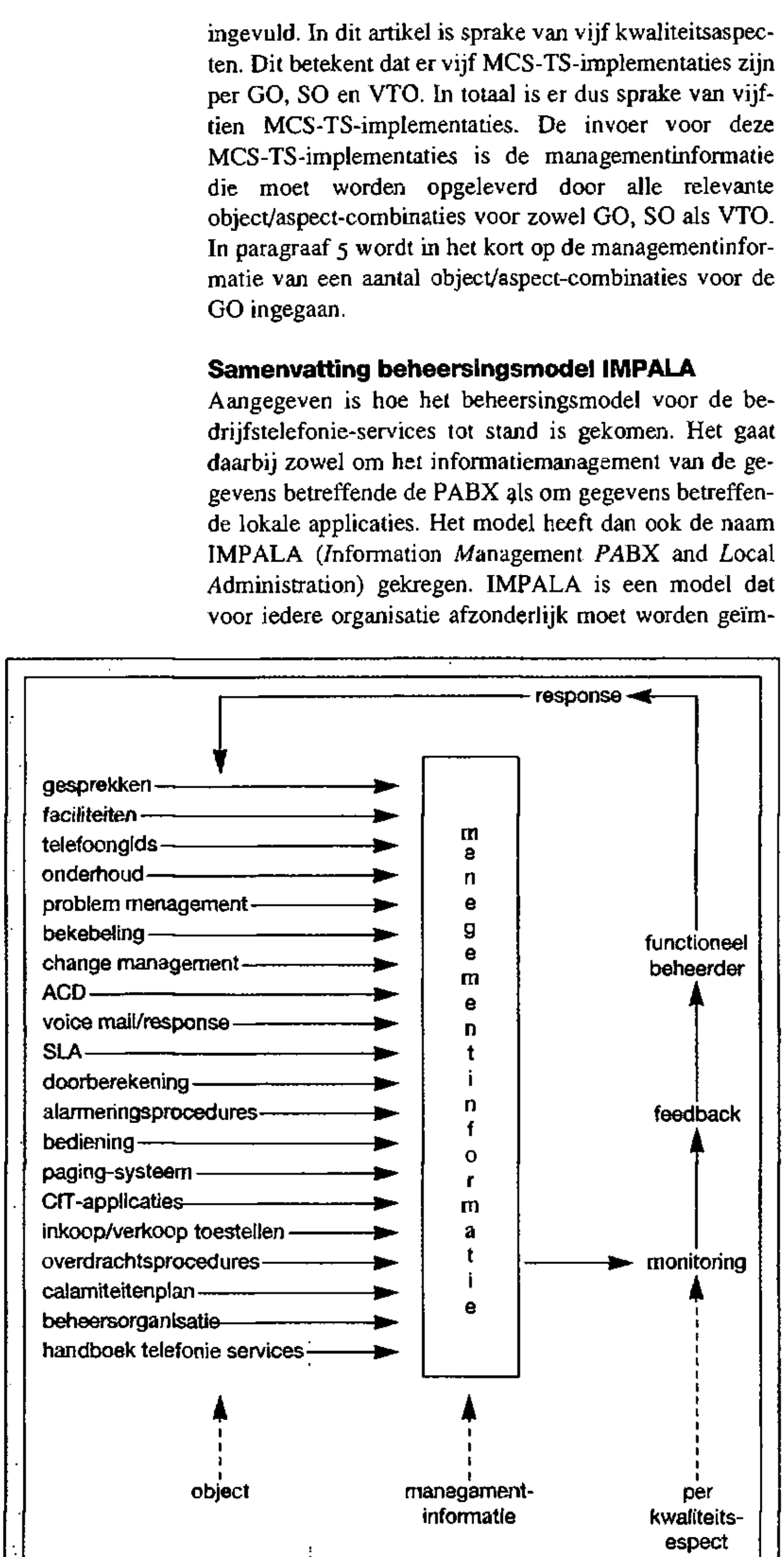
Het MCS-TS moet voor ieder kwaliteitsaspect worden ingevuld. In dit artikel is sprake van vijf kwaliteitsaspecten. Dit betekent dat er vijf MCS-TS-implementaties zijn per GO, SO en VTO. In totaal is er dus sprake van vijftien MCS-TS-implementaties. De invoer voor deze MCS-TS-implementaties is de managementinformatie die moet worden opgeleverd door alle relevante object/aspect-combinaties voor zowel GO, SO als VTO. In paragraaf 5 wordt in het kort op de managementinformatie van een aantal object/aspect-combinaties voor de GO ingegaan.

Samenvatting beheersingsmodel IMPALA

Aangegeven is hoe het beheersingsmodel voor de bedrijfstelefonie-services tot stand is gekomen. Het gaat daarbij zowel om het informatiemanagement van de gegevens betreffende de PABX als om gegevens betreffende de lokale applicaties. Het model heeft dan ook de naam IMPALA (*I*nformation *M*anagement *PA*BX and *L*ocal *A*dministration) gekregen. IMPALA is een model dat voor iedere organisatie afzonderlijk moet worden geïmplementeerd. Het uitwerken van alle relevante TS-object/aspect/organisatie-combinaties is geen sinecure. In de praktijk is het dan ook verstandig om prioriteiten te stellen.

Toepassen van het IMPALA-model betekent het uitvoeren van de volgende stappen:

- *Stap 1:* Stel vast wat de afhankelijkheid van de telefonie-services is vanuit de GO.
- *Stap 2:* Stel vast welke algemene beheersmaatregelen getroffen zijn en of die voldoende zijn voor de telefonie-services.
- *Stap 3:* Stel vast welke TS-objecten van toepassing zijn.
- *Stap 4:* Stel vast welke kwaliteitsaspecten van toepassing zijn.
- *Stap 5:* Bepaal de prioriteit van de kwaliteitsaspecten.
- *Stap 6:* Bepaal de prioriteit van de TS-object/aspect/organisatie-combinaties.
- *Stap 7:* Werk volgens prioriteit alle relevante TS-object/aspect/organisatie-combinaties uit:
  - *a* vul het kwaliteitsaspect in;
  - *b* beschrijf de toepassingsgerichte beheersmaatregelen;
  - *c* bepaal de noodzakelijke managementinformatie.
- *Stap 8:* Richt per kwaliteitsaspect een Management Control System in.



Figuur 5: Management Control System Telefonie Services TU Delft

Op deze manier kan volgens prioriteit worden gewerkt aan het uitwerken van de relevante TS-object/aspect/organisatie-combinaties (stap 7) en wordt het Management Control System voor de telefonie-services (MCS-TS) steeds beter (de stappen 7 en 8 worden iteratief uitgevoerd).

5 Invulling model IMPALA voor de TU Delft

In paragraaf 4 is de beheersingsproblematiek uitgewerkt en is het model IMPALA geconstrueerd. Dit als raamwerk gepresenteerde model moet in de praktijk worden uitgewerkt. Dat betekent dat alle TS-objecten en alle relevante TS-object/aspect-combinaties verder moeten worden gedetailleerd. In deze paragraaf worden enkele object/aspect-combinaties beknopt beschreven.

Object/aspect-combinaties

Effectiviteit van de faciliteiten

Onder effectiviteit van een (informatie-)voorziening kan worden verstaan: 'de mate waarin de faciliteiten *aansluiten bij de verwachtingen* van de gebruikers' (NIVRA, 1989).

Eén van de geboden faciliteiten betreft de groepsschakelingen. Tijdens de inventarisatieronde bij de TU Delft heeft een aantal gebruikers gevraagd om een zogenaamde distributiegroepsschakeling. Zo'n schakeling betekent dat een aantal telefoonnummers (= toestellen) in een *distributiegroep* wordt opgenomen, waarbij de betreffende distributiegroep van een eigen aankiesnummer (groepsnummer) wordt voorzien. Gesprekken die binnenkomen via het groepsnummer, worden nu gelijkelijk *(wrap around)* verdeeld over de toestellen die tot de distributiegroep behoren. Vanuit VTO-standpunt bekeken was de effectiviteit optimaal. Alle distributiegroepen werkten in technisch opzicht perfect. Sommige gebruikers echter waren ontevreden over het resultaat. Ondanks een uitgebreide voorlichtingscampagne hadden ze andere verwachtingen. Pas toen de nieuwe telefooncentrale in gebruik was genomen, bleek dat de door henzelf gekozen oplossing niet effectief was, omdat in de praktijk bleek dat hij niet voldeed aan *hun* verwachtingen.
De betreffende gebruikers zijn bezocht door personeelsleden van 'Informatiebeheer Telefonie Services'. Deze brachten eerst de werk- en communicatiesituatie in beeld en adviseerden vervolgens over de, voor hun situatie geschikte, groepsschakelingen. De gebruikers zijn inmiddels zeer tevreden over de effectiviteit van hun groepsschakelingen.

Managementinformatie
Regelmatig, bijvoorbeeld maandelijks, dient aan het management te worden gerapporteerd inzake de distributiegroepsschakelingen. Daarbij valt te denken aan:
- aantal distributiegroepsschakelingen aan het begin en aan het eind van de maand;
- aantal nieuwe distributiegroepsschakelingen;
- aantal vervallen distributiegroepsschakelingen, met daarbij een korte impressie van de motieven.

Effectiviteit voice mail/response
De belangstelling van toestelgebruikers voor een postbusmogelijkheid is groot. Onder de effectiviteit van voice mail kan worden verstaan: 'voorzien in de mogelijkheid boodschappen voor toestelgebruikers te laten deponeren in een beveiligde postbus en de betreffende toestelgebruikers te attenderen op de aanwezigheid van één of meer boodschappen'.
In verband met de grote belangstelling is het aantal lijnen naar de voice mail-eenheid uitgebreid, zodat geen boodschappen verloren gaan. Toestelgebruikers worden geattendeerd op de aanwezigheid van één of meer boodschappen doordat het message waiting-lampje op het toestel gaat branden als er een boodschap in de betreffende postbus is ingesproken. Daarnaast heeft de toestelgebruiker de mogelijkheid aan zijn postbus een telefoonnummer te koppelen. Indien er nu een boodschap in de postbus is ingesproken, dan zal de PABX een verbinding tot stand brengen tussen de voice mail-eenheid en het opgegeven nummer (bijvoorbeeld het privé-telefoonnummer). De toestelgebruiker kan nu met behulp van zijn pincode de boodschappen in zijn postbus afluisteren.

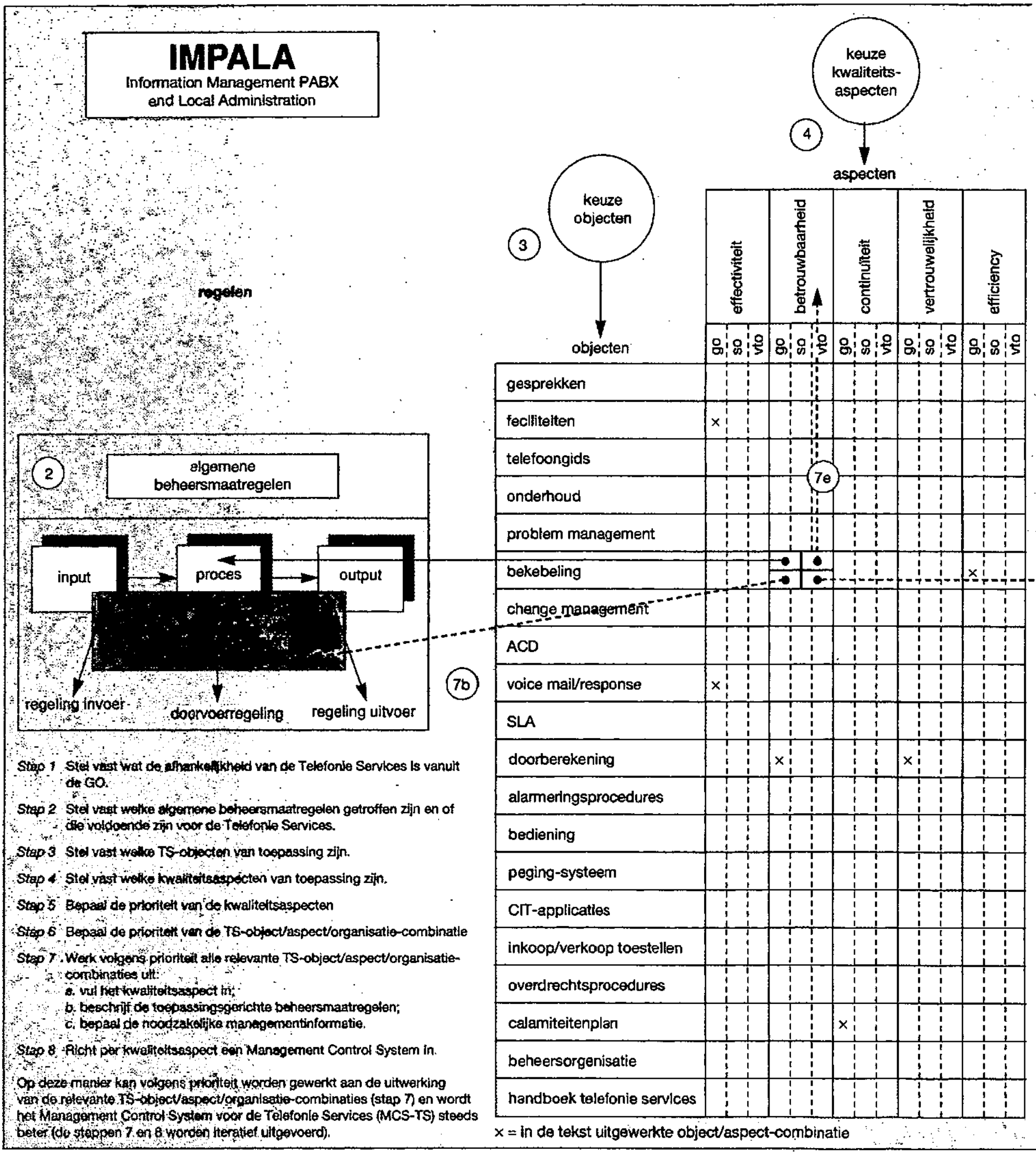
Managementinformatie
Het management ontvangt overzichten omtrent het gebruik van de voice mail-eenheid. Het betreft informatie over het aantal ingesproken boodschappen, het aantal uitgelezen boodschappen, doorbelfrequentie, bezettingsgraad, enzovoort.

Betrouwbaarheid van de doorberekening
Onder betrouwbaarheid kan worden verstaan:
'juistheid, tijdigheid, volledigheid en controleerbaarheid' (EUR, 1992).

Per 1 juli 1994 worden de kosten van de Telefonie Services doorberekend op non-profit basis. Dit betekent onder andere dat de door PTT in rekening gebrachte gesprekskosten worden verbijzonderd en doorberekend aan de juiste gebruikers.
Daartoe is het deelinformatiesysteem 'Doorberekening Telefonie Services (DTS)' ontwikkeld (Van Dijk, 1993). Dit DTS moet voldoen aan bovengenoemde betrouwbaarheidseisen. De controleerbaarheid wordt gegarandeerd door de betreffende budgetbeheerders de mogelijkheid te geven op afroep een overzicht te verkrijgen van door hun organisatorische eenheid (OE) in een bepaalde periode gevoerde gesprekken.
Daartoe zijn alle gesprekken in een audit-trail vastgelegd.

Managementinformatie
De facturen en overige rapportages worden maandelijks verstrekt. Het management ontvangt maandelijks:
- ingedikte informatie over de facturen en overige rapportages aan de organisatorische eenheden;
- een vergelijking over de laatste drie maanden betreffende inkomende (PTT-)rekeningen en verstrekte facturen;
- een vergelijking over de laatste drie maanden van verstrekte facturen met verstrekte facturen in dezelfde perioden in de afgelopen jaren (trendinformatie);
- een storingsoverzicht van het DTS.

Continuïteit van het calamiteitenplan
Continuïteit is te relateren aan drie stadia (Kocks, 1992*b*):
- normaal onderhoud om slijtage van apparatuur en dergelijke tegen te gaan en het treffen van maatrege-



objecten	effectiviteit go	effectiviteit so	effectiviteit vto	betrouwbaarheid go	betrouwbaarheid so	betrouwbaarheid vto	continuïteit go	continuïteit so	continuïteit vto	vertrouwelijkheid go	vertrouwelijkheid so	vertrouwelijkheid vto	efficiency go	efficiency so	efficiency vto
gesprekken															
faciliteiten	×														
telefoongids															
onderhoud															
problem management															
bekabeling													×		
change management															
ACD															
voice mail/response	×														
SLA															
doorberekening				×						×					
alarmeringsprocedures															
bediening															
peging-systeem															
CIT-applicaties															
inkoop/verkoop toestellen															
overdrechtsprocedures															
calamiteitenplan							×								
beheersorganisatie															
handboek telefonie services															

× = in de tekst uitgewerkte object/aspect-combinatie

Stap 1 Stel vast wat de afhankelijkheid van de Telefonie Services is vanuit de GO.

Stap 2 Stel vast welke algemene beheersmaatregelen getroffen zijn en of die voldoende zijn voor de Telefonie Services.

Stap 3 Stel vast welke TS-objecten van toepassing zijn.

Stap 4 Stel vast welke kwaliteitsaspecten van toepassing zijn.

Stap 5 Bepaal de prioriteit van de kwaliteitsaspecten

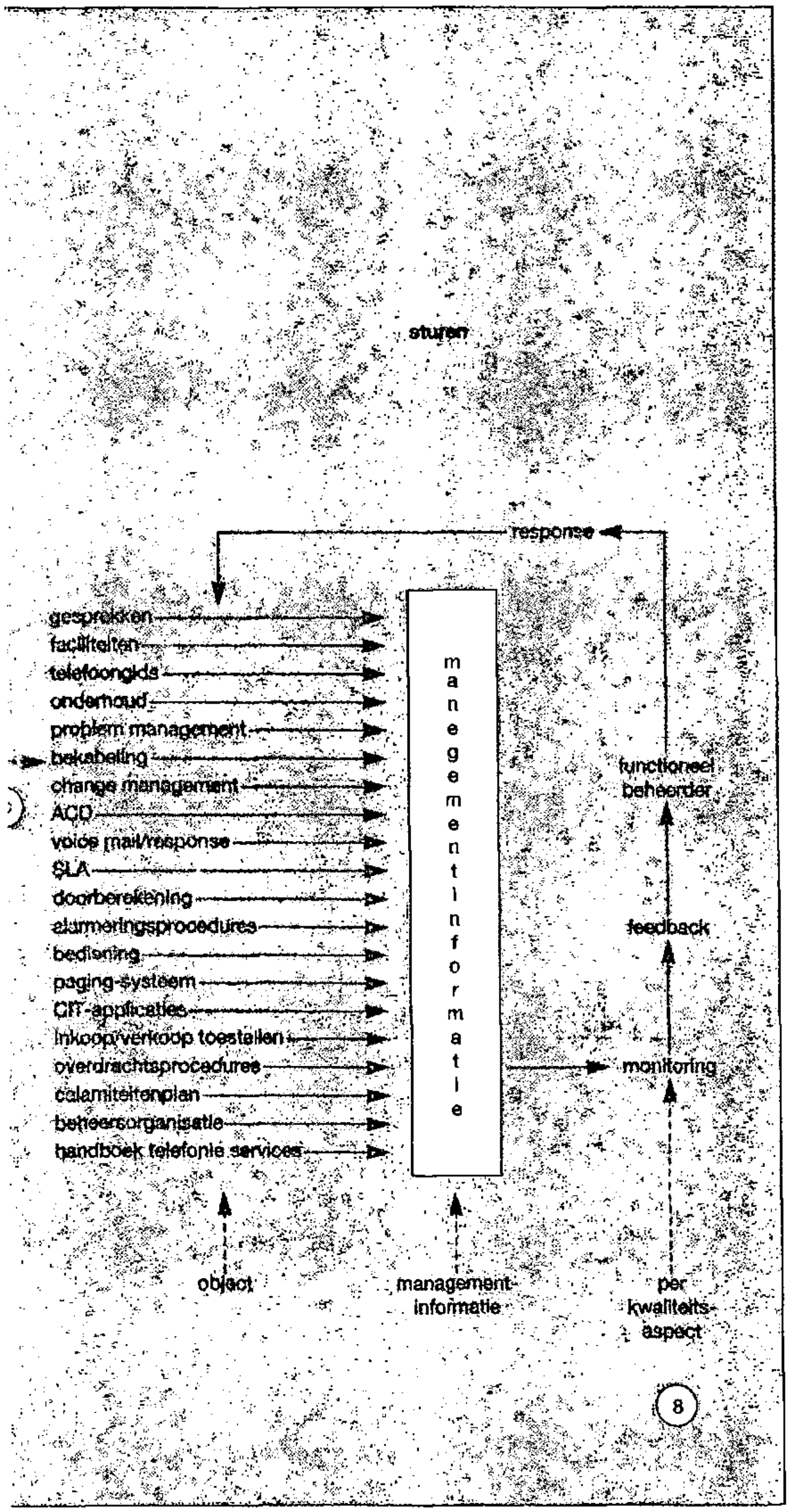
Stap 6 Bepaal de prioriteit van de TS-object/aspect/organisatie-combinatie

Stap 7 Werk volgens prioriteit alle relevante TS-object/aspect/organisatie-combinaties uit:
- *a.* vul het kwaliteitsaspect in;
- *b.* beschrijf de toepassingsgerichte beheersmaatregelen;
- *c.* bepaal de noodzakelijke managementinformatie.

Stap 8 Richt per kwaliteitsaspect een Management Control System in.

Op deze manier kan volgens prioriteit worden gewerkt aan de uitwerking van de relevante TS-object/aspect/organisatie-combinaties (stap 7) en wordt het Management Control System voor de Telefonie Services (MCS-TS) steeds beter (de stappen 7 en 8 worden iteratief uitgevoerd).

Figuur 6: Het beheersingsmodel IMPALA



len opdat de verwerking, bij bijvoorbeeld ziekte, kan doorgaan;
- de beschikbaarheid in het kader van het zich voordoen van een calamiteit dient te zijn gewaarborgd van al die componenten die nodig zijn om te zijner tijd de gegevensverwerking te kunnen voortzetten. Hierbij zijn de termen 'bedrijfseigen' en 'niet-bedrijfseigen' relevant;
- de verwerking dient te kunnen worden hervat binnen een gestelde termijn nadat een calamiteit zich heeft voorgedaan.

Het calamiteitenplan zal mede gebaseerd worden op een risico-analyse, met behulp waarvan de consequenties van een calamiteit in kaart zijn gebracht.

Het minimaliseren van de hersteltijd (één van de doelstellingen) betekent dat er een uitwijkregeling moet zijn getroffen. Deze regeling kan intern of extern (derden) zijn. Er moet een draaiboek zijn waarin het scenario is opgenomen om binnen de gestelde tijd de verwerking te kunnen voortzetten (uitwijkplan). Het draaiboek moet zich vooral richten op de organisatie van het herstellen van de verwerking elders. Het gaat daarbij vooral om de volgende vragen:
- Wie heeft de leiding?
- Wie zijn bij de operatie betrokken?
- Wie (en in welke volgorde) moet worden geïnformeerd?
- De beschrijving en toewijzing van de taken, volgorde van uit te voeren activiteiten (Kocks, 1992*b*).

Het valt buiten het bestek van dit artikel om dieper op het calamiteitenplan in te gaan. Wel kan worden gesteld dat de in het calamiteitenplan opgenomen procedures, voor zover mogelijk, regelmatig moeten worden getest. Belangrijk is ook de continuïteit van het calamiteitenplan zelf. Het moet worden onderhouden en er moeten meer exemplaren van bestaan. Tenminste één van die exemplaren moet worden opgeslagen buiten de gebouwen waar de centrales staan, om te voorkomen dat met een calamiteit ook het calamiteitenplan niet meer beschikbaar is. Ook is het belangrijk bij het herstellen van de verwerking aandacht te besteden aan de overige kwaliteitsaspecten, met name betrouwbaarheid en vertrouwelijkheid.

Managementinformatie
Het management zal over een kopie van het calamiteitenplan moeten beschikken. Het management dient te worden geïnformeerd over wijzigingen in het calamiteitenplan en over de resultaten van testen die worden uitgevoerd in het kader van het calamiteitenplan.

Vertrouwelijkheid van de doorberekening
Het deelinformatiesysteem 'Doorberekening Telefonie Services (DTS)' registreert alle gesprekken die via de telefooncentrale met de 'buitenwacht' worden gevoerd. Bij de gespreksgegevens worden niet alleen het bellende nummer en de gesprekskosten vastgelegd, maar ook het gebelde nummer wordt opgeslagen. Onder 'vertrouwelijkheid met betrekking tot de doorberekening' kan worden verstaan: 'de juiste gebruiker krijgt de juiste gegevens'. In de rapportage-overzichten worden daarom de gebelde nummers niet opgenomen.
In bijzondere gevallen kan informatie worden verstrekt over gebelde nummers. Dit kan alleen via speciale procedures. Ook in die gevallen wordt dan doorgaans volstaan met het verstrekken van de plaats of het land waarnaar gebeld is.

Managementinformatie
Het management ontvangt een overzicht van alle verzoeken om informatie over gebelde nummers en de daarbij gehanteerde afhandelingsprocedures.

Efficiency van de bekabeling
Efficiency kan op een algemeen niveau als volgt worden gedefinieerd: 'De relatie tussen de output die door een systeem wordt voortgebracht en de inputs die nodig zijn geweest om dit mogelijk te maken' (Donkers, Knuvers & Suerink, 1993). De doelstelling van de gebruikersorganisatie kan als volgt worden samengevat: 'Het zo efficiënt mogelijk uitvoeren van de primaire en ondersteunende processen'.

Bij de TU Delft staan thans drie gekoppelde centrales, met daarop aangesloten meer dan 6000 telefoontoestellen, modems, faxen, enzovoort.
Indien er mutaties in het kabelnet moeten worden aangebracht, speelt efficiency een belangrijke rol. De efficiency kan mede worden bereikt door een goed informatiesysteem ten behoeve van het kabelbeheer. Zeker in een omgeving als de TU Delft (groot aantal gebouwen, groot aantal organisatorische eenheden, geografische spreiding) is zo'n voorziening efficiency-bevorderend. Dat is de reden geweest om een grafisch programmatuursysteem voor beheer van het kabelnet aan te schaffen. Binnen de GO kan de efficiency met betrekking tot kabelbeheer extra worden bevorderd door te werken met *patch-panels*.
Daardoor kunnen daartoe bevoegde gebruikers binnen het betreffende (deel van het) gebouw zelf eenvoudige verhuizingen realiseren.

Managementinformatie
Het management moet maandelijks gerapporteerd worden over de aangevraagde mutaties (aantal en soorten). Bovendien moet het management overzichten ontvangen over eventueel te verwachten knelpunten in het kabelnet zoals (bijna) volle verdelers.

6 Samenvatting

De PABX is voor veel bedrijven een belangrijke computer, gezien de afhankelijkheid van de telefonie-services. Het management dient daarom aandacht te hebben voor dit essentiële produktiemiddel. De PABX fungeert ook steeds meer als het centrum van geavanceerde gebruikersnetwerken die tot een groot aantal externe telecommunicatiediensten toegang verschaffen (Ungerer & Costello, 1988). Het beheer van telecommunicatie en datacommunicatie moet op elkaar worden afgestemd. Verwacht mag worden dat veel bedrijven zullen toegroeien naar één systeem voor beheersing van spraak- en datavoorzieningen. De PABX en de daarbij behorende telefonie-services moeten worden voorzien van een beheersingssysteem. Het in dit artikel als raamwerk gepresenteerde model IMPALA kan daarbij als uitgangspunt fungeren en in de praktijk verder worden uitgewerkt.

7 Slotopmerkingen

Dit artikel is gebaseerd op het afstudeerreferaat (Van Dijk, 1994), dat de auteur in mei 1994 heeft afgerond als onderdeel van het slotexamen van de postdoctorale opleiding in EDP-auditing aan de Faculteit der Economische Wetenschappen van de Erasmus Universiteit Rotterdam.

De auteur is dank verschuldigd aan een groot aantal mensen dat het afgelopen jaar bereid was met hem over de onderwerpen 'beheersing' en/of 'bedrijfstelefonie' te discussiëren. Deze discussies hebben bijgedragen aan de 'kwaliteit' van dit artikel. Zijn bijzondere dank gaat uit naar mevr. L.M. Bochove (TU Delft), ir. Ch. Heering (TU Delft), ir. N.P. de Koo (TU Delft) en ir. J.A.M. Donkers (KPMG/EUR).

Literatuur

Bax, A.H.J. (1991), 'PABX', *Handboek Telematica*, april 1991.
Boer, R. de (1992), 'Spraak/data-integratie op applicatieniveau', *Nieuwsbrief Telematica*, oktober 1992.
Brandt, D. e.a. (1993), *In breed verband*, PTT Research, Leidschendam.
Bruijning, J. (1992), 'Informatica van cruciaal belang voor opkomst ISDN', *Computable*, 2 oktober 1992, blz. 7.
Davis, G.B. en M.H. Olson (1985), *Management information systems: Conceptual foundations, structure and development*, McGraw-Hill, New York, 2e druk.
Donkers, J.A.M., P.M. Knuvers, J.M. Suerink (1993), *Collegedictaat GO-Module 4: Efficiëntie*, Erasmus Universiteit Rotterdam.
Dijk, A.J. van (1992), *Definitiestudie Informatiesysteem Telefoongidsbeheer*, Technische Universiteit Delft, juli 1992.
Dijk, A.J. van (1993), *Definitiestudie doorberekening Telefonie Services*, Technische Universiteit Delft, juli 1993.

Afkortingen

AB	applicatiebeheer
ACD	Automatic Call Distribution
CIT	Computer Integrated Telephony
CLI	Calling Line Identification
CSTA	Computer Supported Telephony Applications
DTS	Doorberekening Telefonie Services
EDP	Electronic Data Processing
EUR	Erasmus Universiteit Rotterdam
GB	gegevensbeheer
GO	gebruikersorganisatie
GPT	General Plessey Telecommunications
IMPALA	Information Management PABX and Local Administration
ISDN	Integrated Services Digital Network
ISDX	Integrated Services Digital eXchange
KAD	Kwaliteit van Administratieve Dienstverlening
MCS	Management Control System
OE	Organisatorische Eenheid
OOIV	Ondersteunende Organisatie van de Informatie Voorziening
PABX	Private Automatic Branch eXchange
SLA	Service Level Agreement
SO	organisatie voor systeemontwikkeling en systeemonderhoud
TS	Telefonie Services
TU Delft	Technische Universiteit Delft
VTC	Vervanging Telefooncentrale
VTO	verwerkings- en transporterende organisatie

Dijk, A.J. van (1994), *IMPALA: Een beheersingsmodel voor de bedrijfstelefonie services*, Afstudeerreferaat postdoctorale opleiding EDP-auditing, Erasmus Universiteit Rotterdam, mei 1994.

EUR (1992), *Postdoctorale opleiding in EDP-auditing, Lesbrief (GO) Effectiviteit en Betrouwbaarheid*, Erasmus Universiteit Rotterdam.

EUR (1993), *Postdoctorale opleiding in EDP-auditing*, Erasmus Universiteit Rotterdam.

Hartog, P.A., A. Molenkamp en J.H.M Otten (1992), *Kwaliteit van Administratieve Dienstverlening*, Kluwer Bedrijfswetenschappen, Deventer.

Heck, A.A.J.M. van (1989), *Telefoontechniek compleet*, Estafette, Susteren.

Hendrix, W.H., P. van der Vlist e.a. (1994), *Zaken doen met telematica*, Stichting Teleac, Utrecht.

Hogesteeger, G. (1976), 'De introductie van het fenomeen telefoon in Nederland: 1876 – Het begin van de Telefonie', *Het PTT-Bedrijf*, deel XX, nr. 3, augustus 1976.

Hogesteeger, G. (1989), *Van lopende bode tot telematica: Geschiedenis van de PTT in Nederland*, Koninklijke PTT Nederland NV, Groningen.

Jans, E.O.J. (1989), *Grondslagen van de administratieve organisatie*, Samsom, Alphen aan den Rijn.

Kampert, H.A. e.a. (1991), *Handboek EDP-Auditing: Kwaliteitsbeheersing en kwaliteitsbeoordeling van de geautomatiseerde informatievoorziening*, NIVRA, Amsterdam/Kluwer, Deventer.

Kocks, H.C. (1991), *Inzicht in samenhang*, Collegedictaat EDP-auditing, Erasmus Universiteit Rotterdam.

Kocks, H.C. (1992*a*), *Management Control Oriented Audit Approach*, KPMG-Klynveld EDP Auditors.

Kocks, H.C. (1992*b*), *Continuïteits-audits: Postdoctorale opleiding in EDP-auditing*, Erasmus Universiteit Rotterdam, mei 1992.

Leenhouts, A. (1994), 'Terreur van de telematica: Mobiele communicatie', *Algemeen Dagblad*, 12 maart 1994.

Looijen, M. (1989), *Management en organisatie van automatiseringsmiddelen*, Kluwer Bedrijfswetenschappen, Deventer.

Looijen, M. en G.P.A.J. Delen (1992), *Beheer van informatievoorziening*, Cap Gemini Publishing, Rijswijk.

NIVRA (1989), *Automatisering en controle, Deel VII: Kwaliteitsoordelen over informatievoorziening*, NIVRA-geschrift nr. 53, Kluwer, Deventer.

Oberman, M.R. (1993), *PABX Almanak 1993/1994*, Oberman Telecom Management Consultants BV, maart 1993.

Praat, J. van, en H. Suerink (1992), *Inleiding EDP-auditing*, Kluwer Bedrijfswetenschappen, Deventer.

Selderbeek, J.H. (1992), *Lesbrief Continuïteit: Postdoctorale opleiding in EDP-Auditing*, Erasmus Universiteit Rotterdam, december 1992.

Spaans, E. (1993), 'Telecommunications Management Network (TMN)', *Handboek Telematica*, mei 1993.

Stuvel, H.J. (1952), 'Als de telefoonbel rinkelt', *Natuur en Techniek*.

Sunter, L. (1991), 'De betekenis van CIT en ACD', *Telecommagazine*, nr. 6 (augustus 1991), blz. 40-43.

Systems auditability and control report, Module 10: Contingency planning, The Institute of Internal Auditors Research Foundation, 1991.

Ungerer, H. en N.P. Costello (1988), *Telecommunicatie in Europa*, Serie 'Europese Perspectieven', Brussel.

Verstraaten, E.M.H. (1993), 'ISDN in de praktijk', *Handboek Telematica*, mei 1993.

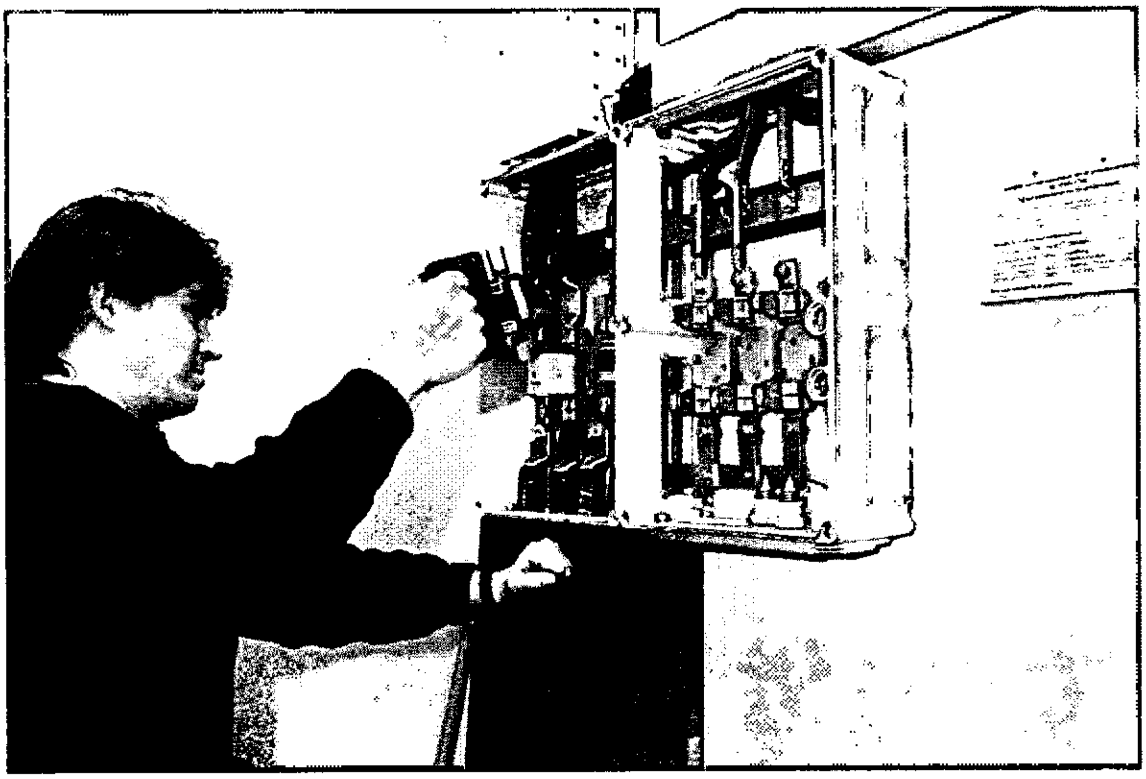
Vonk-Wiersema, G.A. (1990), 'Facsimile', *Handboek Telematica*, januari 1990.

Walters, R. (1993), Computer Telephone Integration, Artech House, Norwood.

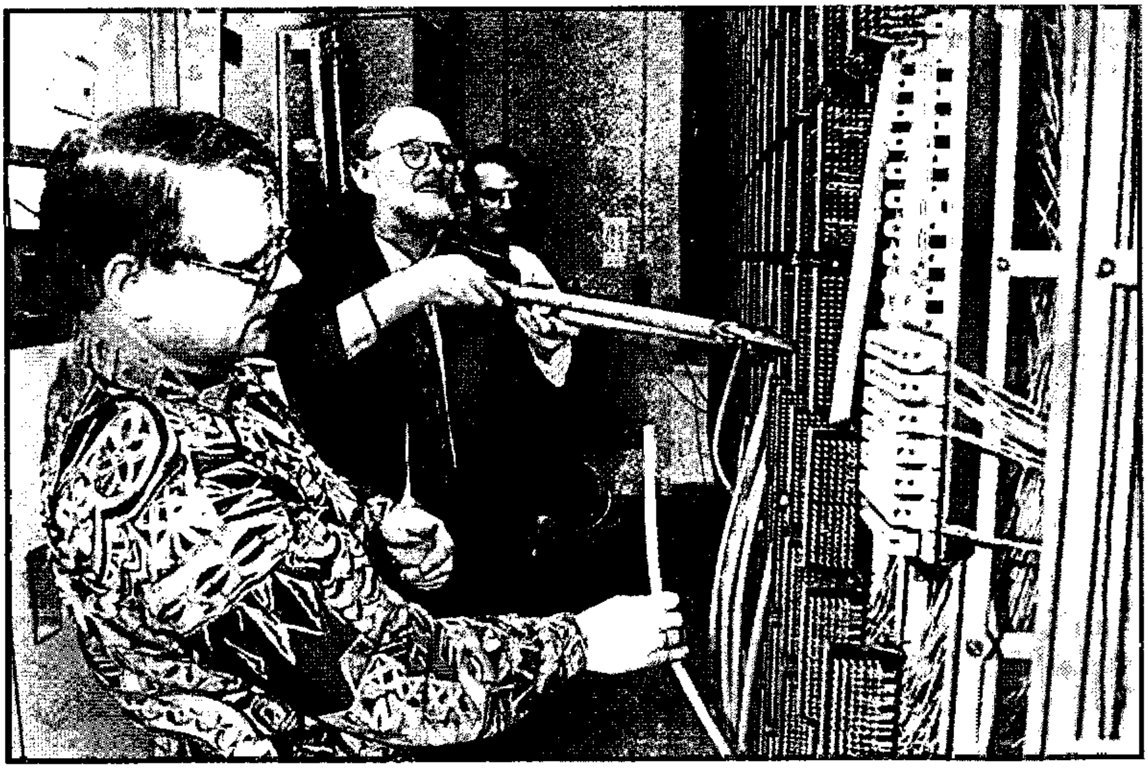
Wijnen, J. van (1993), 'Beter bereikbaar met ACD', *Kantoor & Efficiency*, februari 1993, blz. 25-27.

Ir. Aart J. van Dijk RE RI is werkzaam als zelfstandig informatiekundige/ EDP-auditor (Avédé-Info bv te Zoetermeer). Daarnaast is hij part-time verbonden aan het Rekencentrum van de Technische Universiteit Delft.
Hij is sinds 1965 intensief betrokken bij het ontwerpen, bouwen, invoeren en beoordelen van informatiesystemen.

26 november 1993
Ingebruikneming van de nieuwe centrale

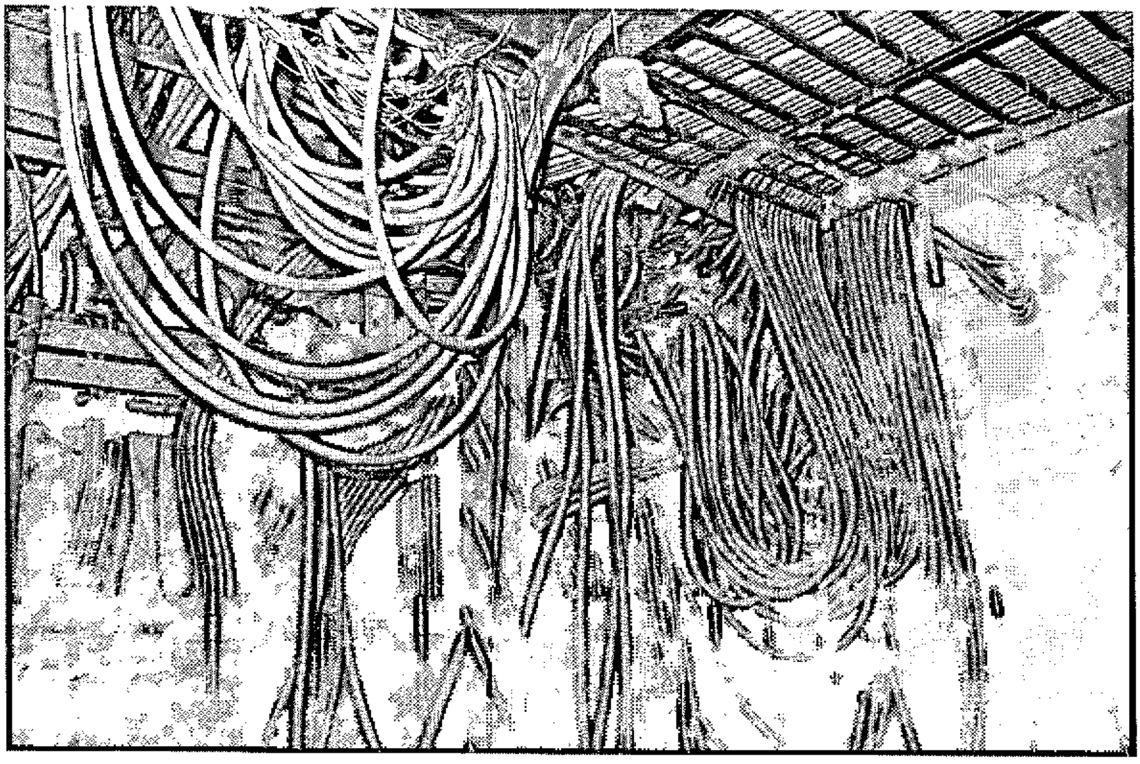


De PTT sluit de oude centrale af

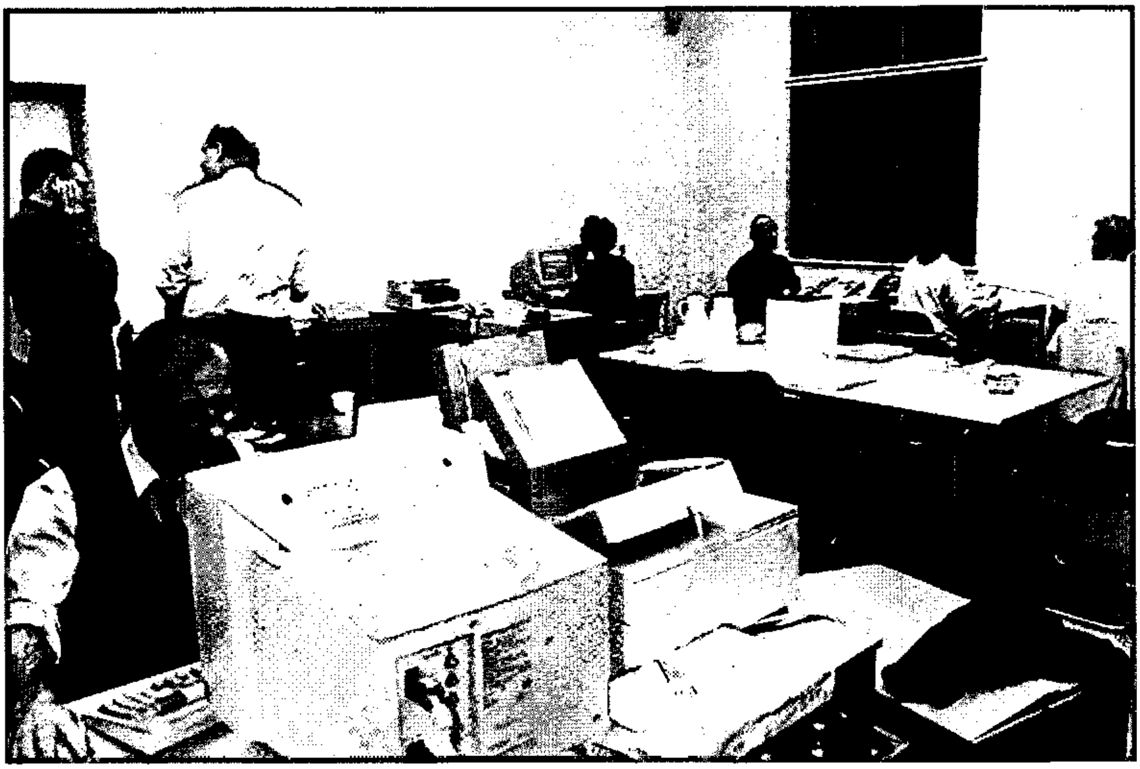


De oude hoofdverdeler wordt "ontkabeld" door ir.J.C. Zuidervaart (direkteur Rekencentrum)

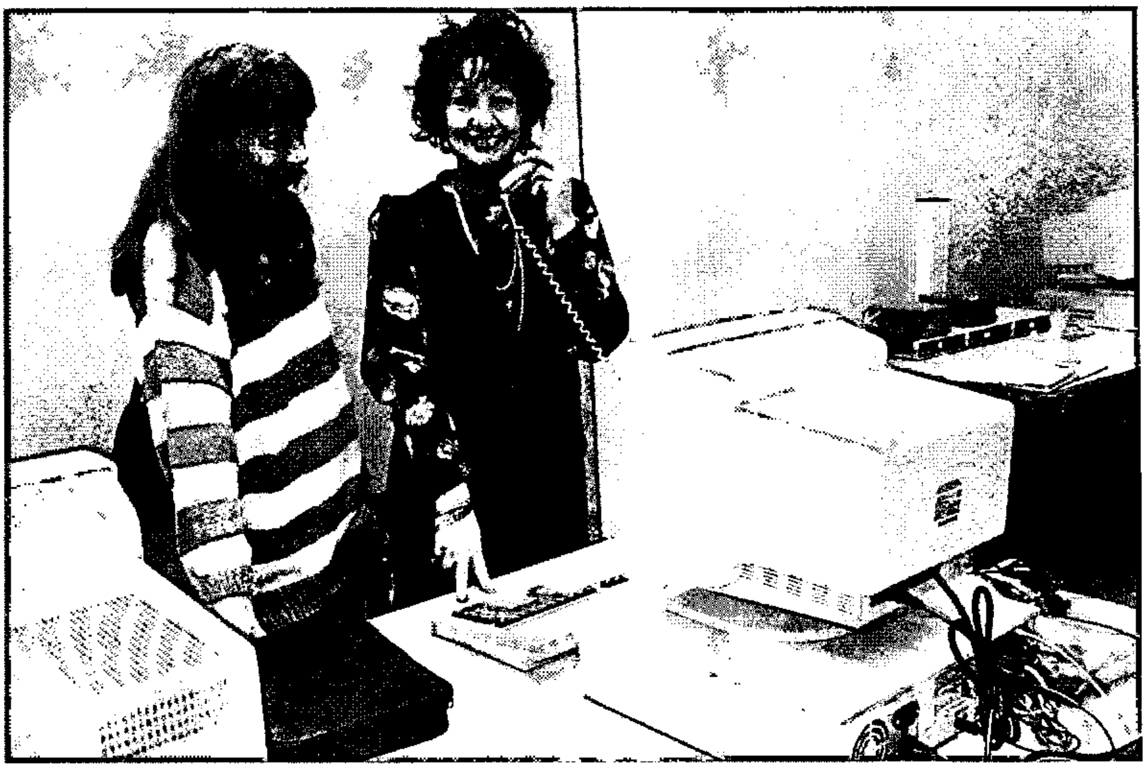
Nachtelijke kabelwerkzaamheden tijdens het overgangsweekeinde



Aan kabels geen gebrek



Het service-centrum tijdens de eerste dagen



Tevreden telefonistes

Publication 8

OKAPI

Schermbeeldcommunicatie en autorisatie

Screen image communication and authorisation

1994

Practising both the trade of advisor as well as auditor
does not in essence need to be conflicting.
It may even be advantageous.
The specific situation dictates whether someone
can and may act as auditor.

Cor Kocks [EUR]

This article previously appeared in:

Informatie, March 1994, pages 191-198

Abstract
In 1994, over 28,000 students were registered at the University of Amsterdam. This university also carries out research. For education and research to run properly, buildings are needed amongst other things. The Dienst Bouw en Huisvesting (DBH) translates and carries out the building and housing policy as decided by the university administration. In 1992, the DBH put the new OKAPI information system into operation. Using OKAPI the building projects are financially supported and monitored. OKAPI was realised using the IDMS DB/DC (network) database management and data communication system and the ADS/ONLINE development tool. During construction the objectives were:

- the users have to be able to navigate quickly and without any problems between the many display images;
- data input and monitoring have to meet very high requirements as far as reliability is concerned;
- users can only carry out authorised actions.

To start with, this article discusses the functionality of OKAPI. Next, it states how the objectives were realised.

Schermbeeldcommunicatie en autorisatie

Aart J. van Dijk

Aan de Universiteit van Amsterdam staan ruim 28.000 studenten ingeschreven. Tevens wordt er onderzoek verricht. Voor een goed verloop van onderzoek en onderwijs zijn onder andere gebouwen nodig. De Dienst Bouw en Huisvesting (DBH) vertaalt en voert het, door het universiteitsbestuur vastgestelde, bouw- en huisvestingsbeleid uit.
De DBH heeft in 1992 het nieuwe informatiesysteem OKAPI in gebruik genomen. Met behulp van OKAPI worden de bouwprojecten financieel begeleid en bewaakt.
OKAPI is gerealiseerd met behulp van het (netwerk-)databasemanagement- en datacommunicatiesysteem IDMS DB/DC en het ontwikkel-tool ADS/ONLINE. Bij de bouw waren de doelstellingen:
- **de gebruikers dienen snel en probleemloos hun weg te vinden tussen de vele schermbeelden;**
- **gegevensinvoer en -bewaking moeten voldoen aan hoge eisen van betrouwbaarheid;**
- **gebruikers kunnen alleen geautoriseerde handelingen verrichten.**

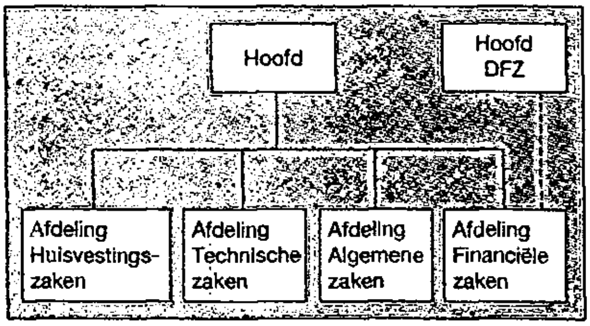
In dit artikel wordt eerst ingegaan op de functionaliteit van OKAPI. Daarna wordt aangegeven hoe de doelstellingen zijn gerealiseerd.

1 Inleiding

De Universiteit van Amsterdam (UvA) is een van de oudste universiteiten in Nederland; het stichtingsjaar was 1632.
15 facultaire beheerseenheden verrichten onderzoek en verzorgen onderwijs voor ruim 28.000 studenten. Daarmee is de UvA een van de grootste Nederlandse universiteiten.
Naast de facultaire beheerseenheden kent de UvA nog enkele andere beheerseenheden, zoals het Bureau van de Universiteit (*Gids UvA*, 1993). Het Bureau bestaat uit de volgende diensten:
- Dienst Onderwijs en Onderzoek;
- Dienst Financiële Zaken (DFZ);
- Dienst Studentenzaken en -welzijn;
- Dienst Personele Zaken;
- Dienst Algemene Zaken;
- Dienst Bouw en Huisvesting (DBH).

De *Dienst Bouw en Huisvesting* (kortweg DBH) heeft tot taak op efficiënte en economische wijze een optimale huisvesting en een adequaat voorzieningenniveau te realiseren voor alle tot de Universiteit van Amsterdam behorende onderdelen, respectievelijk gelieerde instellingen.
De DBH adviseert het universiteitsbestuur omtrent het bouw- en huisvestingsbeleid en voert het ontwikkelde en vastgestelde beleid uit.

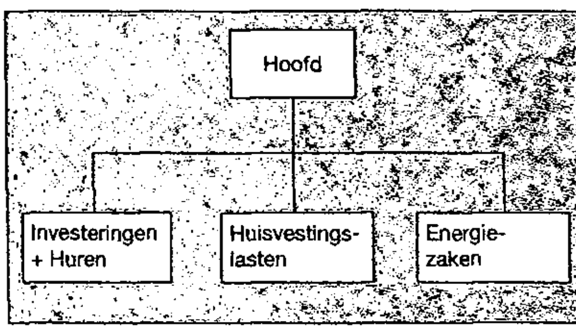
De DBH omvat een viertal afdelingen (zie figuur 1).



Figuur 1: Organisatiestructuurschema DBH

De afdeling *Financiële Zaken* van de DBH draagt zorg voor de financiële beleidsvorming op het gebied van bouw- en huisvestingszaken en biedt ondersteuning bij de financieel-economische, administratieve, procedurele en juridische aspecten.
In het kader van de administratieve organisatie heeft de afdeling Financiële Zaken (zie figuur 2) de volgende taken:
- financiële registratie;
- factuurcontrole;
- toezicht op de rechtmatigheid van de uitgaven;
- begrotingsbewaking;
- secundaire kredietbewaking;
- controle op prijsvormingsprocedures;
- samenstellen van liquiditeitsoverzichten.

De afdeling Financiële Zaken verzorgt onder andere de financiële bouwadministratie, ook wel genoemd 'Kapitaaldienst' (kortweg KPD). In deze administratie worden de nieuwbouw- en verbouwprojecten geadministreerd. De jaarlijkse omzet binnen deze projecten bedraagt enkele tientallen miljoenen guldens.
Ter ondersteuning van deze administratie is in de jaren zeventig een batch-georiënteerd informatiesysteem ontwikkeld (zie *Computersysteem Kapitaaldienst DBH*, 1978). In de jaren tachtig is daaraan een aantal batch-



Figuur 2: Organisatiestructuurschema afdeling Financiële Zaken

programma's toegevoegd. De programmatuur voor KPD draaide op de CDC-computer van de Stichting Academisch Rekencentrum Amsterdam (SARA).
In 1990 besloot de UvA de 'administratieve informatiesystemen' te migreren van de CDC-computer naar de IBM-3090-computer van SARA. Bij de migratie van KPD werd als nevendoel gesteld: 'het actualiseren van de informatiebehoefte'. Een en ander heeft geleid tot een volledige nieuwbouw van het informatiesysteem KPD, waarbij tegelijkertijd diverse kleine informatiesystemen zijn opgenomen. Het nieuwe, geïntegreerde informatiesysteem heeft de naam OKAPI gekregen. Dit acroniem is afgeleid van *On*line *Kapi*taaldienst.

OKAPI is gerealiseerd met behulp van IDMS DB/DC (Integrated Data Management System). Verder is gebruik gemaakt van het ontwikkel-tool 'Application Development System (ADS/ONLINE)'.

In dit artikel gaan wij eerst in op de functionaliteit van de toepassing OKAPI. Daarna wordt aandacht besteed aan de manier waarop de wensen met betrekking tot een snelle, soepele en gebruikersvriendelijke communicatie binnen de 132 schermbeelden van OKAPI zijn gerealiseerd.
Daarnaast komen aan de orde: de manier waarop de gebruikers worden begeleid, de autorisatieregeling binnen OKAPI en de inrichting van het deelsysteem Rapportage. Met dat deelsysteem kan de gebruiker zelf batch-jobs starten en volgen.

2 De toepassing

2.1 Inleiding

In de financiële bouwadministratie worden de nieuwbouwprojecten en de verbouwprojecten geregistreerd. De administratie wordt ondersteund met OKAPI. In de OKAPI-database wordt een groot aantal gegevens geregistreerd: projectgegevens, opdrachten, meerwerk-opdrachten, minderwerk-opdrachten, stelposten, opdrachten uit stelposten, voorlopige opdrachten, termijnen, facturen, betalingen, toegekende budgetten, werkbudgetten, financieringsbronnen, bankgaranties, projectmedewerkers, opdrachtnemers, grootboekrekeningen en kostensoorten (de structuur van de OKAPI-database bestaat uit 36 recordtypen, 42 sets en 16 indexen).
Met behulp van OKAPI worden de bouwprojecten financieel begeleid en bewaakt.
Enerzijds is OKAPI een informatiesysteem voor de financiële bouwadministratie (inclusief budgetbewaking), anderzijds is het een subgrootboekadministratie/crediteurenadministratie.

2.2 Functionaliteit

De DBH heeft niet alleen wensen geuit omtrent de gegevens die in de OKAPI-database moesten worden opgenomen, maar heeft de projectgroep ook voorzien van zeer veel wensen met betrekking tot de wijze waarop OKAPI de processen bij de financiële administratie en de projectleiders moet ondersteunen. De projectgroep is er dankzij een forse inspanning in geslaagd nagenoeg alle wensen te realiseren.
Het on-line-gedeelte van OKAPI bestaat uit een groot aantal schermbeelden, die verdeeld zijn over vijf deelsystemen. De deelsystemen komen in de volgende paragrafen aan de orde. Van ieder deelsysteem zullen enkele aspecten worden besproken.

Deelsysteem 1: Raadplegen/muteren projecten

Deelsysteem 1 bestaat uit 57 schermbeelden en bevat de functies beschrijving projecten, budgetbewaking projecten alsmede de benodigde invoermogelijkheden.
Binnen de functie *beschrijving projecten* kunnen gegevens betreffende projectbeschrijving, projectmedewerkers, financieringsbronnen, financieel jaarverslag en bankgaranties worden opgevraagd en gemuteerd.
Een project bestaat uit een aantal opdracht(sub)groepen. De indeling die thans bij de projecten van de UvA het meest voorkomt, is:
- opdrachtgroep 0: Voorbereidend werk;
- opdrachtgroep 1: Bouwkundige werken;
- opdrachtgroep 2: Technische installaties;
- opdrachtsubgroep 2W: Werktuigbouwkundige installaties;
- opdrachtsubgroep 2E: Elektrotechnische installaties;
- opdrachtsubgroep 2T: Transportinstallaties;
- opdrachtgroep 3: Bijkomende kosten.

De functie *budgetbewaking projecten* is de kern van OKAPI. Binnen deze functie kunnen toegekende budgetten, werkbudgetten, opdrachten, voorlopige opdrachten, facturen, betalingen, enzovoort, worden opgevraagd en gemuteerd.
Enkele entiteiten worden nader toegelicht.

Budgetten
Toegekende budgetten zijn officiële, door de bevoegde instanties goedgekeurde, budgetten die worden toegekend aan een opdracht(sub)groep. Zo kan bijvoorbeeld aan de opdrachtsubgroep 2E van project 276H (verbouw Maagdenhuis) een aantal officiële budgetten worden gekoppeld. Als aan iedere opdracht(sub)groep een aantal officiële budgetten wordt toegekend, ontstaat een (getotaliseerd) budget op drie niveaus: opdrachtsubgroep (bijvoorbeeld 2E), opdrachtgroep (bijvoorbeeld 2) en projectniveau (bijvoorbeeld project 276H).

In de afgelopen jaren bleken projectleiders vaak de wens te koesteren om tijdens de loop van het project met budgetten te kunnen schuiven. Ze wilden dan graag een deel van een budget overhevelen van de ene opdracht-(sub)groep naar een andere. In OKAPI kan naast de officiële budgetten, die worden ingebracht door de financiële administratie, worden gewerkt met werkbudgetten. Deze werkbudgetten kunnen door de betreffende projectleider zelf worden ingebracht. OKAPI controleert of de som van de werkbudgetten van een project niet groter is dan de som van de officiële budgetten.

Indien nu mutaties (zoals opdrachten, facturen, betalingen) worden ingebracht, dan zullen de consequenties in alle zes budgetten worden berekend.

Onderdeel van deelsysteem 1 is het *budgetscherm*. Op dit scherm staan alle budgettaire gegevens van een opdracht(sub)groep, zoals:

- som van de officiële budgetten;
- som van de werkbudgetten;
- restant van de officiële budgetten;
- restant van de werkbudgetten;
- som van de openstaande posten;
- som van de ontvangen facturen;
- som van de betalingen;
- som van de meerwerken;
- som van de minderwerken.

De gebruikers (die daartoe geautoriseerd zijn) kunnen vanaf een groot aantal schermen met behulp van een functietoets direct op dit scherm komen en onmiddellijk de financiële stand van zaken zien, zowel bij de officiële budgetten als bij de werkbudgetten.

Opdrachten

Binnen OKAPI worden verschillende soorten opdrachten onderkend.

De financiële administratie voert de opdrachten in nadat ze een administratieve procedure hebben doorlopen en goedgekeurd zijn. In verband met de verschillende opdrachtsoorten en de relaties daartussen wordt tijdens het invoerproces een groot aantal controles uitgevoerd. De betreffende dialoog kent 49 (fout)meldingen, zoals: 'Foutieve hoofdopdracht', 'Kostensoort onbekend', 'Opdrachtnemer onbekend', 'Foutief BTW-percentage' en 'Meerwerkopdracht bestaat reeds'.

Voorlopige opdrachten

Opdrachten die zijn ingevoerd, hebben een officiële status. Projectleiders hebben vaak, naar aanleiding van discussies op de 'werkvloer', behoefte aan de mogelijkheid snel na te gaan of bepaalde wensen nog binnen het budget kunnen worden uitgevoerd. Daartoe is de entiteit 'voorlopige opdracht' gedefinieerd.

Projectleiders kunnen nu zelf op elk moment voorlopige opdrachten inbrengen, wijzigen en/of verwijderen. Op die manier kunnen ze zelf de budgettaire consequenties van eventuele opdrachten vaststellen.

Op het eerder genoemde budgetscherm kan de gebruiker niet alleen aangeven of hij de financiële stand van zaken bij de officiële budgetten of de werkbudgetten wil zien, maar ook of hij de voorlopige opdrachten wel of niet in de budgettaire beschouwingen wil betrekken.

Deelsysteem 2: Raadplegen/muteren diversen

Deelsysteem 2 bestaat uit 49 schermbeelden.

Met behulp van dit deelsysteem kunnen gegevens worden ingevoerd, opgevraagd en gemuteerd betreffende: gebouwen, BNG-rekeningen (bankrekeningen), kostensoorten, opdrachtgroepen, opdrachtnemers, projectmedewerkers, verzamelposten van betalingen, financieringsbronnen en grootboekrekeningen.

Verzamelposten van betalingen

Betalingen worden ingevoerd in OKAPI en kunnen vervolgens worden afgehandeld via het geautoriseerde financiële systeem FAS van de UvA of via de Bank Nederlandse Gemeenten (BNG). De betalingen worden bundelsgewijs aangeboden aan FAS respectievelijk de BNG. In het laatste geval betreft het betalingen van projecten waarvoor bij de BNG kredieten zijn gedeponeerd. Alvorens een aantal betalingen die een bundel vormen, in OKAPI in te voeren voert men een verzamelpost in. Deze verzamelpost vormt als het ware het label voor de bundel betalingen.

Bij het invoeren van betalingen in OKAPI vindt een groot aantal controles plaats. Allereerst moet er een factuur zijn die betaald kan worden. Deze factuur betreft de gehele of gedeeltelijke termijn van een opdracht.

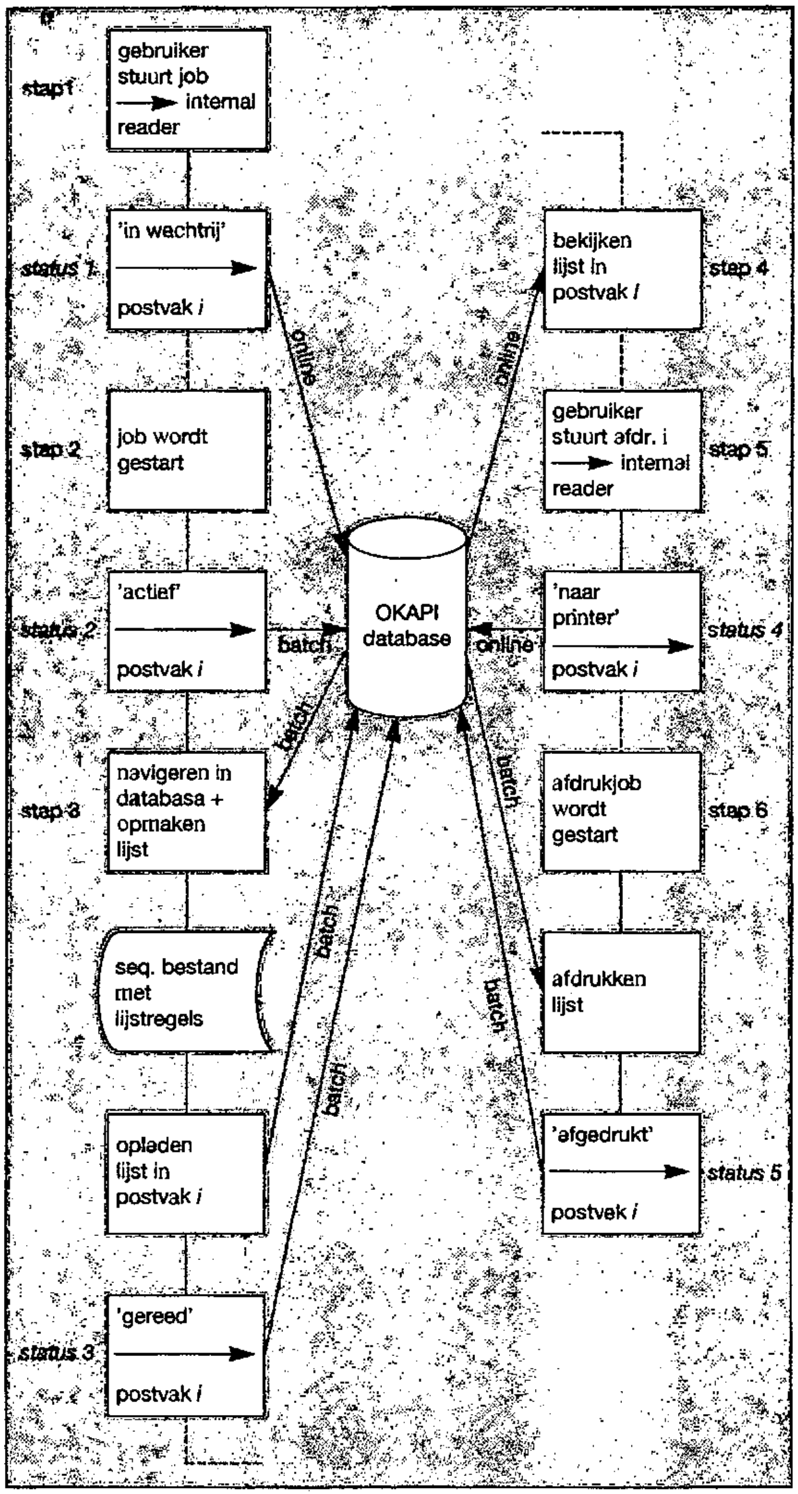
In het geval van een betaling via BNG moet onder andere gecontroleerd worden of het BNG-krediet toereikend is. Bovendien geldt in dat geval op jaarbasis een bovengrens voor de som van de betaalopdrachten.

De dialoog 'invoeren betaling' kent 39 (fout)meldingen.

Deelsysteem 3: Raadplegen crediteuren

Deelsysteem 3 omvat de crediteurenadministratie en bestaat uit 12 schermbeelden. Van een crediteur kunnen gegevens worden opgevraagd betreffende betalingen, openstaande posten, openstaande facturen, volledige opdrachten en facturen waarvoor (nog) geen opdracht geregistreerd is.

Dit laatste komt bijvoorbeeld wel eens voor als de projectleider een meerwerkopdracht heeft verstrekt. Soms vindt de uitvoering van zo'n meerwerkopdracht nog dezelfde dag plaats. De factuur komt dan de volgende dag al binnen. De betreffende opdracht zit echter nog in het invoeringsproces. De financiële administratie heeft er



Figuur 3: Opstarten en verwerken van batch-jobs

behoefte aan dergelijke facturen toch te kunnen invoeren; daarmee beschikt ze over een volledige crediteurenadministratie. Dit is onder andere van belang bij telefonische vragen van opdrachtnemers.

Deelsysteem 4: Rapportage

Gebruikers kunnen met behulp van dit deelsysteem (5 schermbeelden) enkele tientallen parametergestuurde lijsten vervaardigen. Na de keuze van een lijst en het invoeren van de parameters kan de gebruiker een batchjob starten om de lijst te laten vervaardigen. De job wordt door IDMS aangeboden (via de internal reader) en met behulp van de TSO (Time Sharing Option) verwerkt. De resultaten worden binnen de OKAPI-database opgeslagen in een 'postvak' van de betreffende gebruiker, kunnen worden bekeken en desgewenst worden afgedrukt.

Tijdens het vervaardigen van de lijst wordt de gebruiker geïnformeerd over de status van de job. De volgende statussen zijn daarbij mogelijk (zie figuur 3): 'in wachtrij', 'actief', 'gereed', 'naar printer' en 'afgedrukt'. Bij iedere status worden tevens datum en tijdstip vastgelegd. Indien de status 'gereed' wordt bereikt, wordt tevens het aantal uitvoerregels geregistreerd.

Voor de gebruikers van OKAPI voldoet dit aan de informatiebehoefte. Het model kan desgewenst ook voor andere informatiesystemen worden gebruikt. Indien het daarbij wenselijk is de gebruikers nog gedetailleerder te informeren, dan kan het model eenvoudig worden uitgebreid.

Deelsysteem 5: Systeembeheer

Deelsysteem 5 bestaat uit 7 schermbeelden. Met behulp van de functies autorisatie, beginscherm, verstekwaarden kan de functioneel beheerder van OKAPI gebruikers autoriseren en zowel instellingsgegevens als andere verstekwaarden opvragen en muteren.

De wijze waarop de autorisatie is geregeld, komt in de volgende paragraaf aan de orde.

Verstekwaarden kunnen onder andere worden opgegeven voor het actuele dienstjaar, BTW-percentage en grootboekrekening. Het actuele dienstjaar is van belang, omdat het dienstjaar kan afwijken van het kalenderjaar. Met name worden in de maand januari nogal eens facturen ontvangen die betrekking hebben op verrichte werkzaamheden van het jaar ervoor.

Deze facturen en de bijbehorende betalingen worden dan geboekt op het vorig jaar. Zolang de jaarafsluiting niet heeft plaatsgevonden, is dit het actuele dienstjaar.

2.3 Autorisatie

De gebruikers van OKAPI kunnen worden verdeeld in vijf categorieën, te weten:

- de functioneel beheerder van OKAPI (BEH);
- medewerkers van de financiële administratie (FIN);
- het hoofd van de Dienst Bouw en Huisvesting (HBH);
- het hoofd van de afdeling Technische Zaken (HTZ);
- projectmedewerkers/-leiders (PRO).

In een autorisatietabel is aangegeven wie toegang heeft tot welk schermbeeld en wat de bevoegdheden zijn (gebruikersgerichte afscherming, zie Neisingh & Veltman, 1991).
Voor elk schermbeeld is per gebruikerscategorie vastgelegd wat de bevoegdheid is, namelijk
- R = raadplegen is toegestaan;
- M = muteren is toegestaan;
- blank = geen toegang tot het schermbeeld.

Een bijzondere autorisatie is de autorisatie van de categorie van projectmedewerkers. Bij het inbrengen van een nieuw project (door de financiële administratie) wordt verplicht de projectleider ingevoerd. Zodra een project is ingevoerd, moet de betreffende projectleider de autorisatie binnen zijn project zelf regelen. De projectleider kan andere projectmedewerkers autoriseren voor één of meer opdrachtgroepen van zijn project. Op deze manier is er sprake van een tweede autorisatielaag, te weten: een projectgeoriënteerd autorisatieprofiel.
Projectleider en projectmedewerkers moeten vanzelfsprekend binnen OKAPI bekend zijn. Dit is de taak van de functioneel beheerder. Binnen OKAPI is de autorisatie dus gedeeltelijk centraal en gedeeltelijk decentraal geregeld (zie ook Neisingh & Veltman, 1991, blz. 90 e.v.).

2.4 Controles

IDMS DB/DC is te categoriseren als 'proven technology'. De Stichting Academisch Rekencentrum Amsterdam (SARA) is het verwerkingscentrum van de Universiteit van Amsterdam. Niet alleen OKAPI wordt op SARA verwerkt, maar ook het financiële informatiesysteem, het personeelsinformatiesysteem en het studenteninformatiesysteem. SARA heeft de afgelopen jaren bewezen dat er kwalitatief goed werk wordt geleverd. Indien een dergelijke verwerkingsorganisatie (VO) bij voortduring (bewezen) voldoet aan een aantal eisen (zie Kocks, 1991), kan de gebruikersorganisatie (GO) daarop steunen (STeunen OP, ofwel de STOP-situatie).
Indien er sprake is van het beheersingsconcept STOP, dan is de controlefilosofie 'preventief'. In het geval van OKAPI is uitgegaan van een STOP-situatie bij SARA en is uitvoerig aandacht besteed aan controles vooraf.
Dit heeft ertoe geleid dat binnen OKAPI meer dan *800 (fout)meldingen* opgeslagen zijn om de gebruiker te helpen de gegevens correct in te voeren. De combinatie van het beheersingsconcept 'STOP' bij SARA en de controlefilosofie 'preventief' binnen OKAPI maken dat de controles achteraf slechts zeer beperkt behoeven te zijn.

3 Application Structure (ADSA)

ADS/ONLINE
Sinds 1973 is IDMS beschikbaar. In 1979 annonceerde de toenmalige leverancier de TP-monitor IDMS/DC, die geïntegreerd was met IDMS. Omstreeks 1982 kwam het 'ontwikkel-tool' ADS/ONLINE beschikbaar.
ADS/ONLINE moet gezien worden als een TP-toepassing onder IDMS/DC. Centraal staat het begrip dialoog. Een dialoog bestaat doorgaans uit een map (beschrijving schermbeeld), een premap-proces, één of meer responsie-processen, een subschema en werkrecords.
De dialogen worden vervaardigd met behulp van de 'dialog generator' (ADSG). Een dialoog maakt doorgaans deel uit van een verzameling dialogen. Zo'n verzameling dialogen vormt dan een toepassing.
In *Informatie* (november 1984) is een uitgebreid artikel over ADS/ONLINE gepubliceerd (zie Van Dijk, 1984).

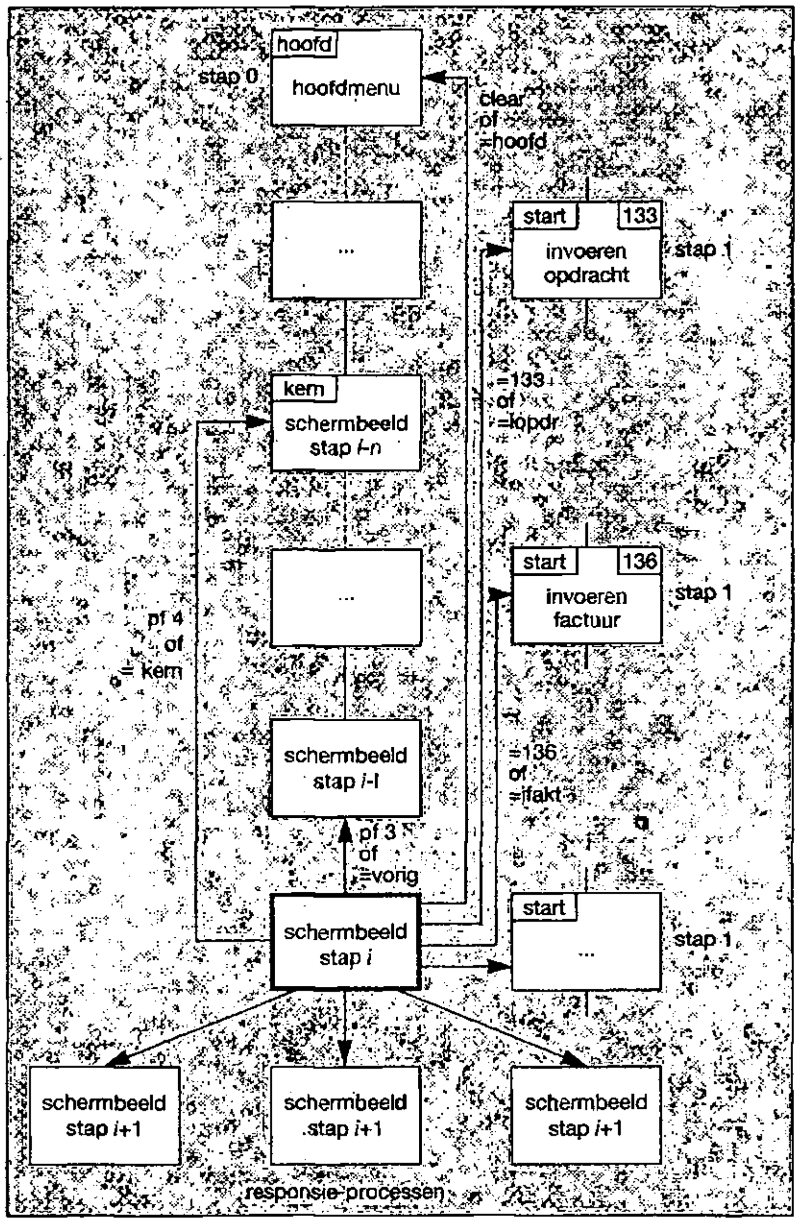
ADSA
In de praktijk sloot de onder ADSG gerealiseerde technische communicatiestructuur nauw aan bij de functionele communicatiestructuur. Het gebruik van veel levels (niveaus in de dialooghiërarchie) binnen ADSG bleek echter nogal bezwaarlijk vanwege het aanzienlijke gebruik van systeemcapaciteit. Langzamerhand gingen de applicatiebouwers dan ook over op een werkwijze waarbij minder niveaus worden gebruikt. Zij werden daarbij geholpen door de komst van de Application Structure Generator (ADSA).
Met ADSA kan, in tegenstelling tot ADSG, de besturing buiten de dialogen worden gehouden en worden overgelaten aan ADSA.

4 ADSA in relatie tot de toepassing

Inleiding
De schermbeelden van OKAPI zijn vastgelegd door middel van Schermbeeld Definities (SD's). De samenhang tussen de schermbeelden is weergegeven met behulp van Schermbeeld-Communicatie-Diagrammen (SCD's; zie Ackema, Van Dijk & Groot, blz. 28 en 29).
Om een snelle, soepele en gebruikersvriendelijke communicatie tussen de schermbeelden te verkrijgen is een aantal eisen vastgesteld.

Eisen aan de schermbeeldcommunicatie
De volgende functionele eisen zijn gesteld aan de schermbeeldcommunicatie (zie figuur 4):

1 Het *pad* dat wordt afgelegd, *moet worden bijgehouden*. Met behulp hiervan moet de terugweg kunnen worden afgelopen. Door het kiezen van de optie 'Vorig niveau' of door gebruik te maken van functietoets 3 wordt één stapje (schermbeeld) teruggegaan van het afgelegde pad (bijvoorbeeld van schermbeeld *i* naar schermbeeld *i*–1).
2 Er is soms behoefte om *'grote stoppen'* terug te gaan langs het afgelegde pad. Dit is het geval als vanaf een bepaald 'kern'-schermbeeld een pad wordt afgelegd. De gebruiker wenst dan met 'één druk op de knop' op dit 'kern'-schermbeeld terug te keren. Op die manier kan de gebruiker bijvoorbeeld snel de financiële consequenties zien van aangebrachte mutaties en/of vanaf het 'kern'-schermbeeld een ander pad 'inslaan' (bijvoorbeeld van schermbeeld *i* naar schermbeeld *i*–*n*).
3 De gebruiker wenst met behulp van de 'Clear-toets' vanaf ieder schermbeeld *terug* te kunnen keren *naar* het schermbeeld *'Hoofdmenu'* (bijvoorbeeld van schermbeeld *i* naar hoofdmenu).
4 Binnen de OKAPI-schermbeelden zijn nogal wat schermbeelden die kunnen fungeren als het begin van een pad. Deze *'start'-schermbeelden* vergen geen andere schermbeelden vooraf en kunnen worden gezien als een nieuwe start binnen OKAPI.
Het is de wens van de gebruiker van *ieder* schermbeeld te kunnen springen naar elk 'start'-schermbeeld (bijvoorbeeld van schermbeeld *i* naar 'invoeren opdracht').
5 Voor sommige schermbeelden is het wenselijk dat ze in *hoofdletters* staan, zodat alle gegevens in hoofdletters worden ingevoerd. Voor andere schermbeelden zijn kleine letters gewenst. Het is daarom gewenst dat de hoofdletter/kleine letter-optie op schermbeeldniveau kan worden geregeld.
In de 'kleine letter-situatie' zullen soms op 'veld'-niveau nog gegevens moeten worden omgezet in hoofdletters.
6 Binnen OKAPI worden verschillende gebruikerscategorieën onderkend. Gebruikers mogen alleen op een schermbeeld komen als ze daartoe geautoriseerd zijn.
7 De gebruiker verlangt dat zoveel mogelijk fouten in de invoergegevens worden gesignaleerd.



Figuur 4: Mogelijkheden van schermbeeldcommunicatie

Als technische eis is gesteld: 'Realiseer OKAPI met behulp van ADSA en streef een technische communicatiestructuur na met een minimaal aantal niveaus'.

De bouw
Uit onderzoek bleek dat niet alle eisen konden worden gerealiseerd binnen het ADSA-concept. Met name het registreren van het afgelegde pad en sprongsgewijs het pad terug aflopen was niet mogelijk.

Om toch zoveel mogelijk gebruik te maken van het ADSA-concept is gekozen voor een oplossing waarbij een eigen 'OKAPI-monitor' werd ingezet. Deze monitor is een map-loze dialoog, die door elke dialoog wordt aangeroepen alvorens de besturing wordt gegeven aan ADSA (zie figuur 5). Met behulp van de OKAPI-monitor zijn alle bovengenoemde eisen gerealiseerd.

Voorbeeld
Een gebruiker staat op het schermbeeld 'Budgetscherm

van opdrachtsubgroep 2E van project 276H'. Dit schermbeeld is een 'kern'-schermbeeld (bijvoorbeeld het schermbeeld stap *i–n* in figuur 4) en laat alle budgettaire gegevens zien van opdrachtsubgroep 2E. De gebruiker doet nu vanaf dit schermbeeld een aantal stappen om opdrachten te bekijken en een mutatie in te brengen (en staat nu op schermbeeld stap *i* in figuur 4). Door nu op de PF4-toets te drukken komt de gebruiker meteen terug op het laatst gebruikte 'kern'-schermbeeld (in dit geval schermbeeld stap *i–n*) en kan hij de budgettaire consequenties van de aangebrachte mutatie beoordelen.

In technische zin gebeurt het volgende. De gebruiker wenst van schermbeeld stap *i* (met het interne dialoognummer 48) te gaan naar schermbeeld stap *i–n* (met het interne dialoognummer 87). Daartoe vinden de volgende vier acties plaats (zie figuur 5):

1 De dialoog met nummer 48 roept de OKAPI-monitor aan met het verzoek het nummer te geven van de laatste 'kern'-dialoog.

2 De OKAPI-monitor bepaalt het gevraagde nummer, controleert de autorisatie, registreert en geeft het gevraagde dialoognummer.

3 De dialoog met nummer 48 roept nu ADSA op met de mededeling dat de volgende dialoog de dialoog met nummer 87 is.

4 ADSA zorgt ervoor dat de dialoog met nummer 87 geactiveerd wordt.

5 Ervaringen

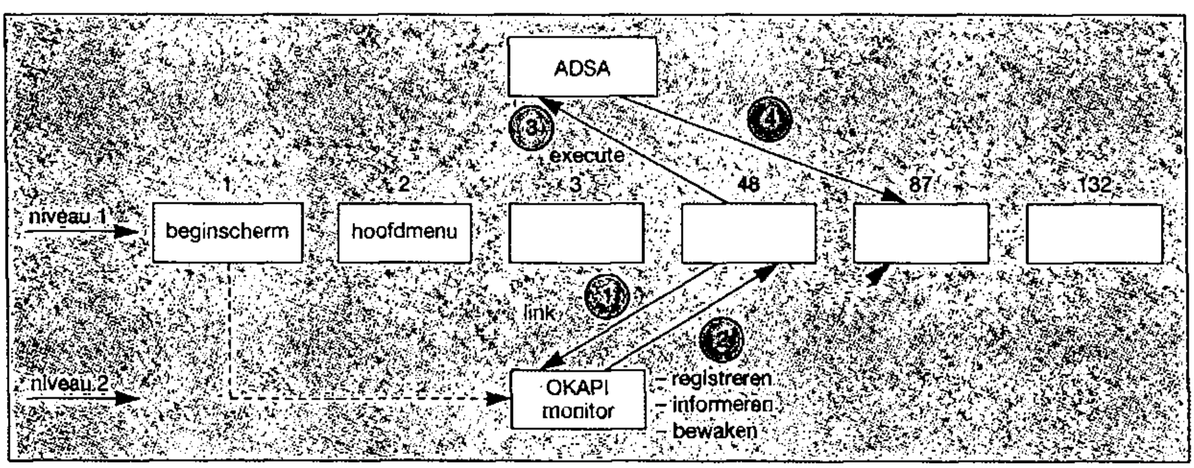
OKAPI wordt door de DBH gebruikt sinds juli 1992. De ervaringen met de communicatie tussen de schermbeelden en het opstarten en volgen van batch-jobs zijn positief. De OKAPI-monitor en het model voor batch-jobs zullen dan ook zeker worden gebruikt bij de bouw van nieuwe toepassingen onder IDMS/ADSA.

6 Projectevaluatie

Het ontwerpen, bouwen en invoeren van informatiesystemen is geen eenvoudige zaak. Regelmatig wordt in publikaties melding gemaakt van mislukte projecten. Ook binnen een pluriforme organisatie als een universiteit lukt het soms niet en soms wel om de realisatie van een informatiesysteem tot een goed einde te brengen.
OKAPI is een voorbeeld van een succesvolle realisatie, hoewel de doorlooptijd wat langer is geweest dan verwacht.

In de periode september 1990-januari 1991 zijn de fasen definitiestudie en functioneel ontwerp uitgevoerd. Het functioneel ontwerp is niet alleen in rapportvorm uitgebracht, maar er is tevens een prototype vervaardigd. Met behulp van dit prototype konden de gebruikers zich een goed beeld vormen van het toekomstige systeem. Tijdens deze fasen hebben de informatici en de belangrijkste gebruikers nauw samengewerkt.
In april 1991 is het functioneel ontwerp aan het management van het Bureau van de Universiteit en de DBH gepresenteerd door middel van een lezing en een demonstratie van het prototype. Deze presentatie werd verzorgd door projectleider en gebruiker.

De projectgroep kreeg vervolgens de opdracht op 1 mei 1991 met de bouw te beginnen. Essentieel daarbij was dat vanaf 1 juli 1992 de CDC-computer, waarop het oude informatiesysteem werd verwerkt, bij SARA niet meer beschikbaar zou zijn. Op basis hiervan is een planning opgesteld. Goede automatiseringsexpertise is ook binnen de Universiteit van Amsterdam slechts beperkt beschikbaar. Naast de part-time projectleider konden 3 ontwerpers/programmeurs worden ingezet. Deze ontwerpers/programmeurs konden echter slechts part-time worden ingezet in verband met andere automatiserings-



Figuur 5: De rol van de OKAPI-monitor in relatie tot ADSA

activiteiten. Ondanks een grote inzet van de betrokkenen was de beschikbare menskracht niet voldoende om het project volledig af te ronden vóór 1 juli 1992. Wel waren op dat tijdstip alle essentiële on-line functies gereed. In de periode juli-december 1992 is onder andere nog gewerkt aan het deelsysteem Rapportage, de jaarwerkprocedures en het afronden van de documentatie. Tijdens de bouw is ook veel tijd besteed aan het deelproject Conversie. Naast conversieprogrammatuur om de IDMS-database in de laatste week van juni 1992 te kunnen laden, is een aantal programma's gemaakt om de oude gegevensbestanden te kunnen analyseren en waar nodig te corrigeren.

Samenvattend kunnen de belangrijkste personele kosten en baten als volgt worden weergegeven.

Personele kosten

Ten behoeve van de fasen definitiestudie en functioneel ontwerp waren 4 mensmaanden informaticus en 1 mensmaand gebruiker nodig. Voor de bouw en invoering van OKAPI moesten 40 mensmaanden informaticus en 6 mensmaanden gebruiker worden ingezet.

Baten

De Dienst Bouw en Huisvesting van de Universiteit van Amsterdam beschikt over een omvangrijk (maatwerk-) informatiesysteem waarmee de komende tien jaar de nieuwbouwprojecten en de verbouwprojecten, die vermoedelijk enkele honderden miljoenen aan investeringen zullen vergen, op een efficiënte en effectieve manier financieel begeleid en bewaakt kunnen worden.

OKAPI wordt niet alleen gebruikt door de financiële medewerkers, ook de projectleiders van de bouwprojecten worden in hun dagelijkse werkzaamheden ondersteund en het management wordt tijdig voorzien van de juiste informatie.

Literatuur

Ackema, H., A.J. van Dijk en H. Groot (1991), *Functioneel ontwerp OKAPI (Online Kapitaaldienst)*, Universiteit van Amsterdam, Bureau der Universiteit, Memorandum no. UvA-BIA-91002, januari 1991, 101 blz.

Afdeling Automatisering (1978), *Computersysteem Kapitaaldienst DBH*, Universiteit van Amsterdam, afdeling Automatisering, juni 1978.

Application Development System/Online, Features guide.

Application Development System/Online, Reference guide.

Dijk, A.J. van (1984), 'ADS/ONLINE, Ervaringen met een "ontwikkeltool" bij de ontwikkeling van het information retrieval systeem GIRAF, *Informatie*, jaargang 26, nr. 11, blz. 912-927.

Eilers, H.B. (1989), *Systeemontwikkeling op kleine schaal met SDM*, Academic Service, Schoonhoven (ISBN 90 6233 456 3).

Esseling, E.K.C. en H. van Nimwegen (1992), *Administratieve processen*, Kluwer Bedrijfswetenschappen, Deventer, 266 blz.

Kocks, H.C. (1991), *Inzicht in Samenhang*, Collegedictaat EDP-auditing, Erasmus Universiteit Rotterdam, 133 blz.

IDMS/R, Concept and facilities.

Neisingh, A.W. en P. Veltman (1991), *Ontwikkelingen informatietechnologie: Risicobeheersing en beveiligingsaspecten*, KPMG Klynveld EDP auditors, Amsterdam, 111 blz.

Universiteit van Amsterdam, *Gids Universiteit van Amsterdam 1993-1994*.

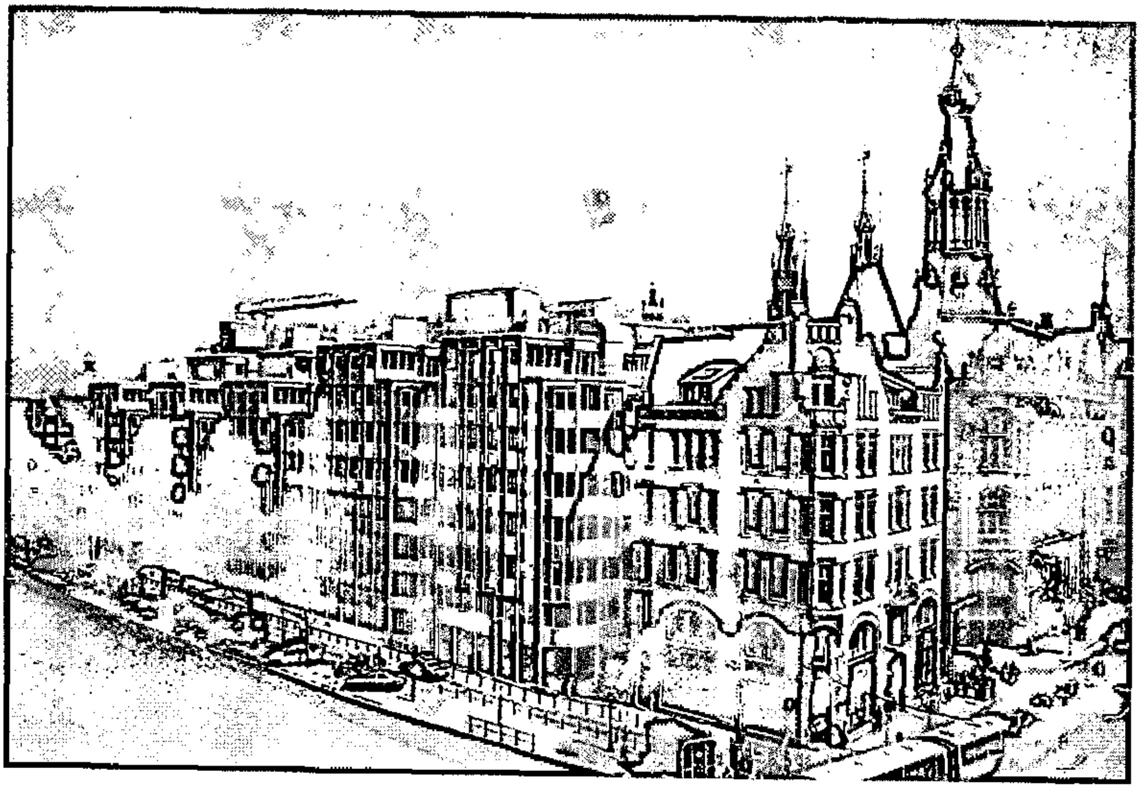
OKAPI is ontworpen en gebouwd door Hans Ackema, Carlo Busman, Aart van Dijk (projectleider), Henk Groot en Laura Martens. De projectgroep heeft gerapporteerd aan een stuurgroep, die bestond uit J. Keppel (Hoofd afdeling Bestuurlijke Informatie en Automatisering) en G.J. Haak (Hoofd afdeling Financiële Zaken van de Dienst Bouw en Huisvesting).

Ir. Aart J. van Dijk RE RI is werkzaam als zelfstandig informatiekundige/EDP-auditor (Avédé-Info bv te Zoetermeer).

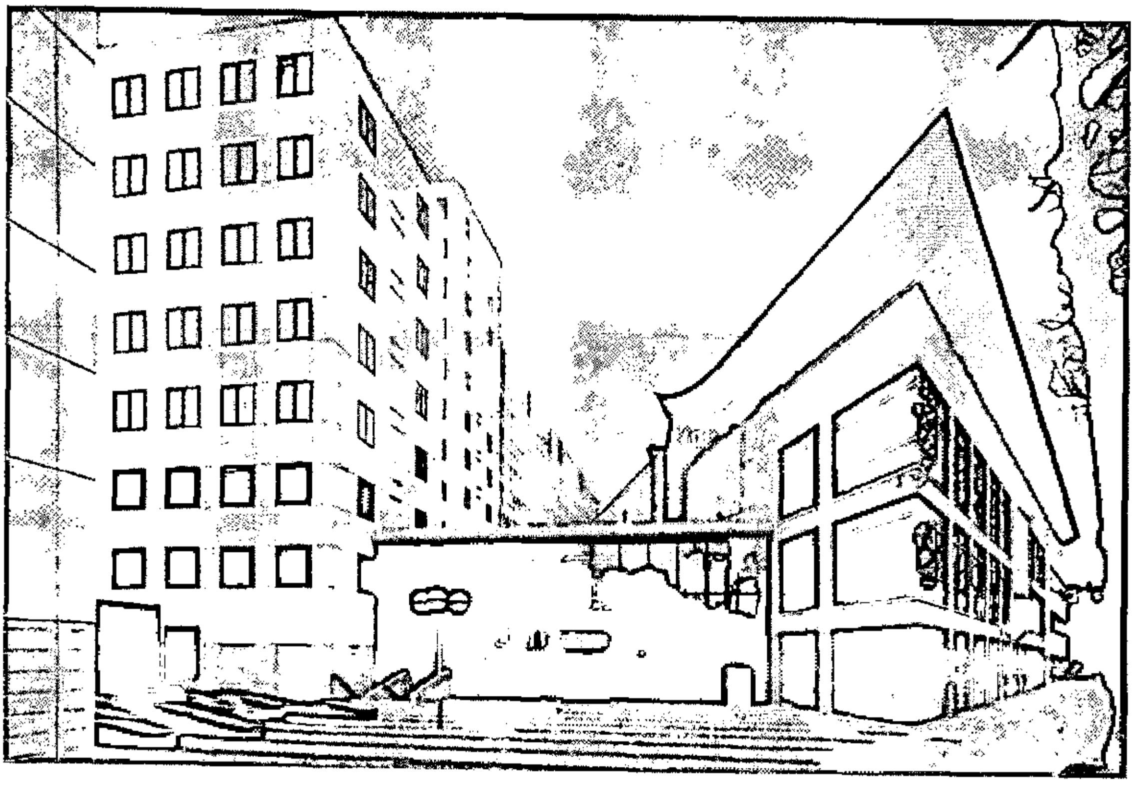
Daarnaast is hij part-time verbonden aan het Rekencentrum van de Technische Universiteit Delft.

Hij is sinds 1965 intensief betrokken bij het ontwerpen, bouwen, invoeren en beoordelen van informatiesystemen. De Universiteit van Amsterdam heeft hem gedurende de periode 1991-1992 aangetrokken als externe projectleider voor het OKAPI-project.

Produkten van de Dienst Bouw en Huisvesting



Faculteit der Letteren



Roeterseiland

Studenten Informatie Centrum

Publication 9

GIRAF

Een information retrieval systeem voor universiteiten, hogescholen en

andere pluriforme organisaties

An information retrieval system for Universities,
Schools of Higher Vocational Education
and other multiform organisations

1984

Poor project management will defeat good engineering,
and is the most frequent cause of project failure.

Watts Humphrey [1998]

This article previously appeared in:

Informatie, May 1984, pages 445-455

Abstract
In 1970, the Delft University of Technology (DUT) developed the 'BIBLIOSYSTEM' (batch) system. Information on this system was published in the Dutch magazine 'Informatie' of October 1971. Although the system was intended to make the composition of bibliographies using the computer run faster and more smoothly, the system was also used hoth inside as well as outside the DUT in the period 1970-1984 for processing all types of files such as directories, membership files, hardware files, book files, files with data on conference participants, etc. In 1976, the Computer Centre of the DUT developed the online retrieval system 'BIBINFO'. This system fitted in with 'BIBLIOSYSTEM'. The constant demand within the DUT for similar facilities resulted in development of the, according to the 1984 standards, user-friendly information retrieval system 'GIRAF'. At realisation of 'GIRAF', the IDMS DB/DC database management and data communication system acted as a support system. Furthermore, the development tool 'Application Development System (ADS/ONLINE)', which is supplied by the same supplier as IDMS DB/DC, was used. This article goes into the functional aspects of 'GIRAF'.

GIRAF

EEN INFORMATION RETRIEVAL SYSTEEM VOOR UNIVERSITEITEN, HOGESCHOLEN EN ANDERE PLURIFORME ORGANISATIES

door ir. A. J. van Dijk

In 1970 is bij de Technische Hogeschool Delft het (batch-)systeem 'BIBLIOSYSTEM' ontwikkeld. Over dit systeem is gepubliceerd in het blad 'Informatie' van oktober 1971. Hoewel het systeem bedoeld was om het samenstellen van bibliografieën met behulp van de computer sneller en prettiger te laten verlopen, is het in de periode 1970-1984 zowel binnen als buiten de TH-Delft ook gebruikt voor het bewerken van allerhande bestanden zoals adresbestanden, ledenbestanden, apparatuurbestanden, diabestanden, boekbestanden, bestanden met gegevens van congresgangers, etc. In 1976 ontwikkelde het Rekencentrum van de TH-Delft het on-line retrieval systeem 'BIBINFO'. Dit systeem sloot aan op 'BIBLIOSYSTEM'. De voortdurende vraag binnen de TH-Delft naar dergelijke faciliteiten heeft er toe geleid dat het, naar de normen van 1984, gebruiksvriendelijke information retrieval systeem 'GIRAF' is ontwikkeld. Bij de realisatie van 'GIRAF' heeft het database-management- en data-communicatie-systeem IDMS DB/DC als draagsysteem gefungeerd. Bovendien werd gebruik gemaakt van het ontwikkeltool 'Application Development System (ADS/ONLINE)' dat, evenals IDMS DB/DC, door de firma Cullinet Software geleverd wordt.
Dit artikel behandelt de functionele aspecten van 'GIRAF'.

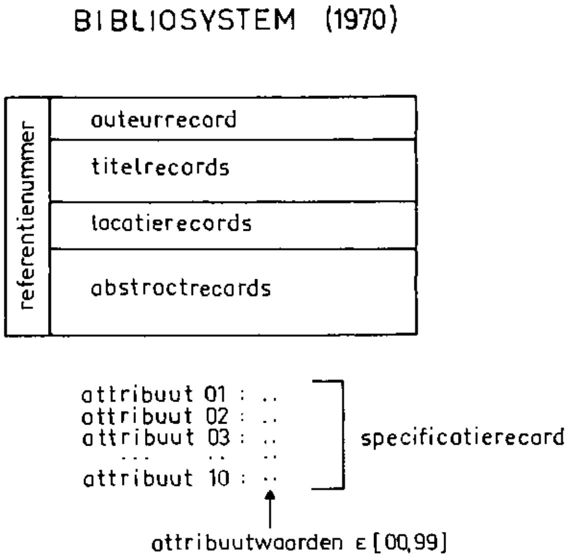
1 INLEIDING

In 1970 is bij de Technische Hogeschool Delft (kortweg TH-Delft) het systeem 'BIBLIOSYSTEM' [3] ontwikkeld. Aanleiding voor de bouw van dit systeem was de grote hoeveelheid tijd, die onderzoekers moesten besteden aan het verzamelen, registreren en ordenen van literatuur. Het bestuderen van literatuur kreeg nl. een steeds belangrijkere plaats in het onderzoek en het samenstellen van bibliografieën en literatuurlijsten werd onderdeel van vrijwel ieder researchproject van enige betekenis. Een grote flexibiliteit in het bijwerken en classificeren bleek daarbij van grote waarde.
De klassieke wijze van rapporteren van een literatuurstudie was zeer tijdrovend o.a. vanwege de omvang van het typewerk. Dit typewerk vormde een ware bottleneck. Bibliografieën waren soms op het moment dat ze gereed waren voor verspreiding alweer verouderd. Een en ander leidde tot het idee om het samenstellen van bibliografieën te automatiseren. 'BIBLIOSYSTEM' werd begin 1971 door de TH-Delft in gebruik genomen. Binnen de TH-Delft bleek het in een grote behoefte te voorzien. Maar niet alleen binnen de TH-Delft was men geïnteresseerd; na een publikatie in het blad 'Informatie' (oktober 1971) [3] kreeg de TH-Delft meer dan honderd verzoeken om de systeembeschrijving en programma's beschikbaar te stellen. In de loop der jaren bleek bovendien, dat door de 'abstracte' opzet het systeem bruikbaar was voor allerhande toepassingen. Naast bestanden met literatuurreferenties (zie figuur 2) werden met behulp van 'BIBLIOSYSTEM' adresbestanden, bestanden voor ledenadministraties, apparatuurbestanden, diabestanden, boekbestanden, etc. vervaardigd, bijgewerkt en afgedrukt.
In 1975 werd het Rekencentrum geconfronteerd met het verzoek van een congresorganisatiecomité om tijdens een te organiseren congres in de zomer van 1976 een realtime congresinformatiesysteem beschikbaar te stellen. Dit leidde tot de ontwikkeling van 'BIBINFO' [4]. Dit systeem werd als het ware bovenop 'BIBLIOSYSTEM' gebouwd. Het opbouwen van de bestanden gebeurde met 'BIBLIOSYSTEM'. Het raadplegen en het on-line selecteren gebeurde met 'BIBINFO'. In 1976 was het systeem voor de congresorganisatie beschikbaar. In de periode 1976-1983 is 'BIBINFO' regelmatig gebruikt door de 'BIBLIOSYSTEM'-gebruikers en door congresorganisaties.
'BIBLIOSYSTEM' (1970) en 'BIBINFO' (1976) zijn ontwikkeld voor een beperkte doelgroep. De systemen zijn ontwikkeld met methoden en hulpmiddelen, die in die jaren modern waren. Het grote succes van deze systemen binnen de TH-Delft en de voortdurende vraag naar dergelijke faciliteiten heeft de directie van het Rekencentrum in 1982 doen besluiten een nieuw information retrieval systeem te laten ontwikkelen, dat op een ruimere doelgroep moest worden gericht. Het moest de systemen 'BIBLIOSYSTEM' en 'BIBINFO' vervangen en zeer gebruiksvriendelijk zijn. Dit nieuwe systeem is het *programmatuursysteem* 'GIRAF' (*G*eneral *I*nformation *R*etriev*A*l *F*acilities) geworden.
In dit artikel zullen de *functionele* aspecten van 'GIRAF' worden behandeld. De technische aspecten komen slechts zeer zijdelings ter sprake. In een volgend artikel zal worden ingegaan op enkele technische aspecten van 'GIRAF'. Bovendien zal in dat artikel worden ingegaan op de ervaringen opgedaan met conversationele programmering ('BIBINFO' is volledig conversationeel geprogrammeerd), pseudo-conversationele programmering [5] en programmering met behulp van een 'ontwikkeltool' ('GIRAF' is ontwikkeld met behulp van het ont-

wikkeltool 'Application Development System (ADS/ ONLINE)' van de firma Cullinet Software [1, 2]).

2 UITGANGSPUNTEN 'GIRAF'

Uit het succes van de systemen 'BIBLIOSYSTEM' en 'BIBINFO' is de conclusie getrokken, dat de filosofie, waarop deze systemen waren gebaseerd, voor de TH-Delft een goede filosofie is. De eerste voorwaarde, die aan 'GIRAF' is gesteld, is het handhaven van deze filosofie. Dit betekende o.a., dat 'GIRAF' een eenvoudig en doeltreffend information retrieval systeem moest worden (*eis 1*).
De gegevenseenheid in de oude systemen was de referentie (zie figuren 1 en 2). Toegang tot een *individuele*



Figuur 1: Gegevensmodel van een referentie binnen 'BIBLIOSYSTEM' (1970)

referentie geschiedde via het referentienummer of de auteur. Toegang tot een *verzameling* referenties was alleen mogelijk via een codesysteem. Dit codesysteem bestond uit een tiental attributen met voor ieder attribuut een waardenverzameling. Aan iedere referentie kon per attribuut een waarde uit de betreffende waardenverzameling worden toegekend. Op deze manier kon een gebruiker aan een referentie kenmerken en kenmerkwaarden toekennen. Aan een boekreferentie kon bijvoorbeeld in attribuutnummer 2 de waarde 38 en in attribuutnummer 8 de waarde 17 worden toegekend (zie figuur 2). Een en ander betekende dat het boek in de Engelse taal was geschreven en door een bepaalde uitgever was uitgegeven. Het selecteren van een deelverzameling was te realiseren door de attributen, waarin men voor de betreffende selectie was geïnteresseerd, op te geven. Per attribuut werd de waardenverzameling, die voor deze selectie geldig was, opgegeven. Met behulp van EN- en OF-relaties op attribuutniveau was het mogelijk de gewenste deelverzameling te construeren.
Aangezien 'GIRAF' voor een ruimere doelgroep moest worden gemaakt, moest het meer algemeen bruikbaar zijn dan de oude systemen (*eis 2*). Daaruit werd, mede naar aanleiding van verzoeken van gebruikers, de conclusie getrokken, dat het codesysteem ruimer zou moeten zijn (*eis 3*).
De TH-Delft is een pluriforme organisatie, die veel zelfstandig opererende groeperingen herbergt. Sommige van die groeperingen willen (delen van) hun informatievoorziening op een geïsoleerde manier realiseren (bijv. promovendi) (*eis 4*). Andere groeperingen willen graag samenwerken met groeperingen binnen of buiten de TH-Delft. Voor die groeperingen is het noodzakelijk om vanuit verschillende locaties, met behulp van real-time-verwerking, bestanden te kunnen opbouwen, raadplegen en/of muteren (*eis 5*).
'GIRAF' moet kunnen worden toegepast ten behoeve van veel informatiesystemen binnen de TH-Delft. Daarom moet 'GIRAF' worden aangeboden als *confectiesysteem*. De opzet moest echter dusdanig zijn, dat het met weinig inspanning kan worden aangepast. Op die manier kan 'GIRAF' voor sommige gebruikers als *maatconfectiesysteem* fungeren (*eis 6*).
Bij de TH-Delft werken en studeren ook buitenlanders. Het is daarom wenselijk dat 'GIRAF' kan werken met andere talen. Het systeem moest zodanig worden opgezet, dat in een later stadium *andere taalversies* snel kunnen worden geïmplementeerd (*eis 7*).
Het Rekencentrum van de TH-Delft is een dienstverlenende instantie. Zoals het een goede dienstverlenende

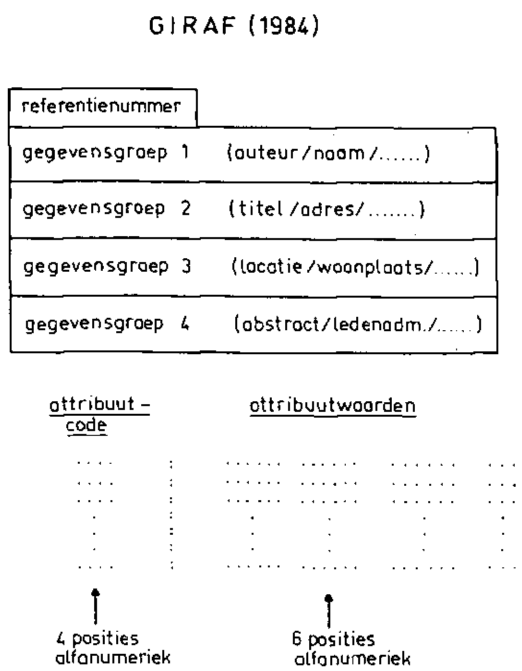
```
45682711 5010 Koenen,M.J. en J. Endepols                      (auteurrecord)
45682711 6010 Verklarend handwoordenboek der Nederlandse taal (titelrecord)
45682711 7010 Vooral ten dienste van het onderwijs            (locatierecord)
45682711 8010 Zesentwintigste druk                            (abstractrecord 1)
45682711 8020 Uitgeverij : Wolters-Noordhoff NV  Groningen    (abstractrecord 2)

Attribuut  1 : 47     (type boek)
Attribuut  2 : 38     (taal van het boek)
Attribuut  3 : 00
Attribuut  4 : 00
Attribuut  5 : 00                                              (specificatierecord)
Attribuut  6 : 02     (bij wie in de kast)
Attribuut  7 : 00
Attribuut  8 : 17     (uitgever)
Attribuut  9 : 00
Attribuut 10 : 00
```

Figuur 2: Voorbeeld van een 'BIBLIOSYSTEM'-referentie

instantie betaamt, geeft zij een hoge prioriteit aan de belangen van haar huidige gebruikers. 'GIRAF' diende daarom voorzieningen te hebben om de gebruikers van 'BIBLIOSYSTEM' en 'BIBINFO' op een zeer soepele wijze te kunnen laten overstappen naar het nieuwe systeem (*eis 8*).
Door het Rekencentrum van de TH-Delft is als draagsysteem ten beboeve van database-management- en datacommunicatie-activiteiten gekozen voor IDMS DB/DC, dat wordt geleverd door de firma Cullinet Software. 'GIRAF' diende met behulp van IDMS DB/DC te worden ontwikkeld (*eis 9*). 'GIRAF' kan daardoor in principe functioneren op elke computer waarop een recente versie van IDMS DB/DC is geïnstalleerd. Vaak zal dat een computer uit de serie IBM 4300 zijn, maar ook andere computers kunnen werken met IDMS DB/DC. Het Rekencentrum van de TH-Delft heeft 'GIRAF' operationeel op een Amdahl 470V/7B computer.

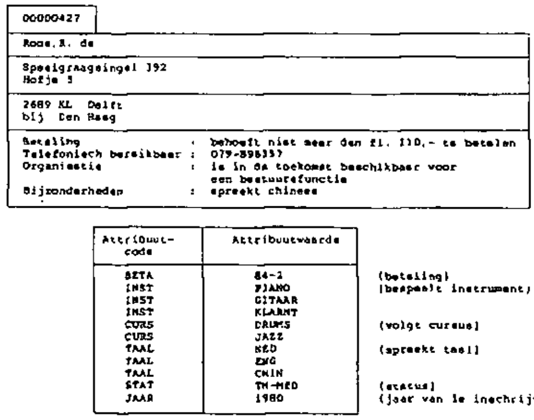
3 INDELING VAN DE REFERENTIES

De eenheid in de systemen 'BIBLIOSYSTEM' en 'BIBINFO' was de referentie. Hoewel oorspronkelijk bedoeld voor het behandelen van literatuurreferenties (zie figuur 2) werden de systemen voor allerhande toepassingen gebruikt. De naamgeving was echter gebaseerd op literatuurreferenties. Zo was er sprake van een auteur, een titel, een locatie en een abstract. Daarnaast werd aan een referentie voor een aantal (maximaal 10) attributen een attribuutwaarde toegekend. Aangezien de term 'referentie' bij veel gebruikers van de oude systemen is ingeburgerd, is besloten deze term in 'GIRAF' over te nemen. 'GIRAF' moest een algemeen karakter hebben, daarom kon niet worden volstaan met de bestaande naamgeving. Binnen 'GIRAF' bestaat een referentie uit vier gegevensgroepen (zie figuur 3). Iedere gebruiker die verantwoordelijk is voor een gegevensverzameling (in



Figuur 3: Gegevensmodel van een referentie binnen 'GIRAF' (1984)

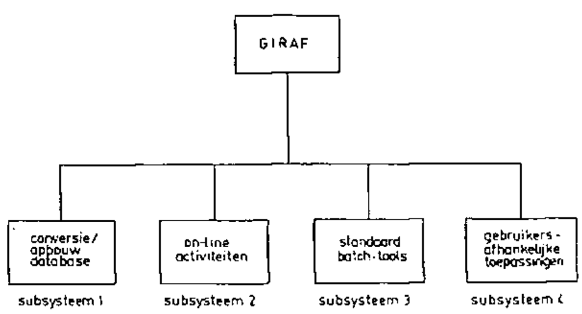
'GIRAF'-termen: iedere ToepassingsBestandsBeheerder (TBB)) kan zelf de namen kiezen, die hij aan deze gegevensgroepen wenst te geven. Overal waar de gegevensgroepen worden gebruikt (op beeldschermen of papier) worden de door de TBB opgegeven namen gebruikt. 'GIRAF' kent zelf aan iedere nieuwe referentie een referentienummer toe. Het codesysteem, waarmee attribuutcodes en attribuutwaarden aan een referentie kunnen worden toegekend, is uitgebreid (zie figuren 3 en 4). Het aantal attribuutcodes is nu onbeperkt. Bovendien kunnen per referentie aan een attribuutcode verschillende alfanumerieke attribuutwaarden worden toegekend.



Figuur 4: Voorbeeld van een 'GIRAF'-referentie

4 SUBSYSTEMEN

Het programmatuursysteem 'GIRAF' is opgedeeld in 4 subsystemen (zie figuur 5), te weten:
subsysteem 1: conversie/opbouw database
subsysteem 2: on-line activiteiten
subsysteem 3: standaard batch-tools
subsysteem 4: gebruikersafhankelijke toepassingen
Allereerst worden nu de subsystemen 1, 3 en 4 kort toegelicht. Daarna komt subsysteem 2 aan de orde.



Figuur 5: Subsystemen 'GIRAF'

4.1 Conversie/opbouw database (subsysteem I)

Met behulp van dit subsysteem kunnen bestanden in 'BIBLIOSYSTEM'-formaat worden geconverteerd naar een 'GIRAF-Toepassings-Database' (GTD). Met behulp van stuurinformatie kan tot op zekere hoogte rekening worden gehouden met individuele conversiewensen. Gebruikers die in het bezit zijn van een gegevensverzameling, die niet met 'BIBLIOSYSTEM' is opgebouwd, kunnen via subsysteem 4 toch overstappen naar 'GIRAF'.

4.2 **Standaard batch-tools (subsysteem 3)**
Gebruikers van 'GIRAF' krijgen door middel van subsysteem 3 een aantal gereedschappen voor batch-verwerking. Zo bevat dit subsysteem uitgebreide *rapportagefaciliteiten,* zoals statistische overzichten omtrent de omvang en samenstelling van de GTD, het afdrukken van een gebruikershandleiding, het afdrukken van een attribuutcodeoverzicht etc. Essentieel voor het maken van selecties (deelbestanden) is het codesysteem met attribuutcodes en attribuutwaarden. Het selecteren van referenties op basis van teksten in de gegevensgroepen 2, 3 en 4 (zie figuur 3) is on-line niet mogelijk. Desondanks kan het voorkomen, dat een gebruiker op basis van een *inhoudelijke tekst* referenties wil selecteren. Tot de standaard batch-tools behoort een programma, waarmee het mogelijk is om dergelijke selecties te maken. Met behulp van dit programma wordt de betreffende GTD geheel doorzocht. Een dergelijke verwerking kan dan ook alleen 's nachts plaatsvinden en moet een uitzondering blijven. Het selecteren van referenties via subsysteem 2 (on-line activiteiten) is gebonden aan een aantal restricties. Eén van deze restricties is bijvoorbeeld de omvang van het deelbestand dat wordt geconstrueerd. Het deelbestand mag maximaal 1000 referenties bevatten. Het on-line bestuderen van meer dan 1000 referenties wordt niet zinvol geacht. Wenst een gebruiker grotere deelbestanden te construeren, dan kan hij dit realiseren door gebruik te maken van een *batch-selectieprogramma.* Verder bevat dit subsysteem o.a. programma's om referenties *opnieuw in* te *delen* en programma's om een GTD van *codesysteem* te laten *veranderen.*

4.3 **Gebruikersafhankelijke toepassingen (subsysteem 4)**
'GIRAF' wordt aangeboden als *confectiesysteem.* De opzet is echter dusdanig, dat met weinig inspanning wijzigingen kunnen worden aangebracht. Op die manier kan van een *maatconfectiesysteem* worden gesproken. Bepaalde categorieën van gebruikers zullen 'GIRAF' willen (laten) uitbreiden met een 'eigen' on-line systeem. De structuur van 'GIRAF' maakt dit mogelijk. Het eigen on-line systeem zal aan bepaalde randvoorwaarden moeten voldoen, en kan op een eenvoudige manier binnen 'GIRAF' worden opgenomen. Het eerste gebruikersafhankelijke subsysteem binnen 'GIRAF', dat ontwikkeld is, is een subsysteem voor de CentraleBestandsBebeerder (CBB) van 'GIRAF'. Deze CBB heeft een aantal gereedschappen om het centrale beheer over de gegevensverzamelingen te kunnen voeren en om, in het geval dat zich problemen voordoen, gebruikers te kunnen assisteren. Desgewenst kunnen eigen batch-tools voor gebruikers worden ontwikkeld, zoals bijvoorbeeld specifieke conversieprogrammatuur. Met behulp van deze programmatuur kan een elders opgebouwde gegevensverzameling worden geconverteerd naar een GIRAF-Toepassings-Database (GTD).

4.4 **On-line activiteiten (subsysteem 2)**
Het on-line subsysteem bestaat uit een zevental functies (zie figuur 6). Indien voor de betreffende gebruiker een (on-line) gebruikersafhankelijk subsysteem beschikbaar is, dan kan via functie 7 toegang tot dit systeem worden verkregen. Naast deze functies is er een functie, die de toelating tot een GTD regelt.
Allereerst wordt nu ingegaan op deze toelatingsprocedure. Daarna zullen de belangrijkste andere functies in het kort worden behandeld.

4.4.1 *Toelatingsprocedure*
Iedere beheerder van een GTD (ToepassingsBestands-Beheerder (TBB)) stelt in overleg met de Centrale BestandsBeheerder (CBB) een zogenaamd *systeemkenmerk* en een tijdelijk *wachtwoord* vast. Vanaf dat mo-

```
Beeld 0.3.1

*****************************************************************************
* *** TECHNISCHE HOGESCHOOL DELFT ***                                       *
* *** GIRAF    : "Bestands-ident" ***                                       *
*                                                                           *
*                                                                           *
*                      *** FUNKTIEMENU ***                                  *
*                                                                           *
*                                                                           *
*                  *1*   HELP (HANDLEIDING)                                 *
*                  *2*   OPHALEN/MUTEREN INDIVIDUELE REFERENTIE             *
*                  *3*   SELECTEREN/VERWIJDEREN VERZAMELING REFERENTIES     *
*                  *4*   OPSTARTEN SERIE-VERWERKING (BATCH)                 *
*                  *5*   INBRENGEN NIEUWE REFERENTIE(S)                     *
*                  *6*   WIJZIGEN BESTANDSGEGEVENS                          *
*                 "*7*   Gebruikersafhankelijke funktie"                    *
*                  *9*   AFSLUITEN SESSIE                                   *
*                                                                           *
*                  MAAK UW FUNKTIE-KEUZE KENBAAR DOOR HET                   *
*                  BETREFFENDE GETAL IN TE TOETSEN : _....                  *
*                                                                           *
*                  U KUNT  DE  GEKOZEN FUNKTIE TE  ALLEN TIJDE              *
*                  VERLATEN DOOR OP DE "CLEAR"-TOETS TE DRUKKEN             *
*                                                                           *
* -.-.-.-.-.-.-.-.-.-.-.-.-.-"systeem-boodschappen"-.-.-.-.-.-.-.-.-.-.-.-.-.*
*****************************************************************************
```

Figuur 6: Functiemenu van 'GIRAF'

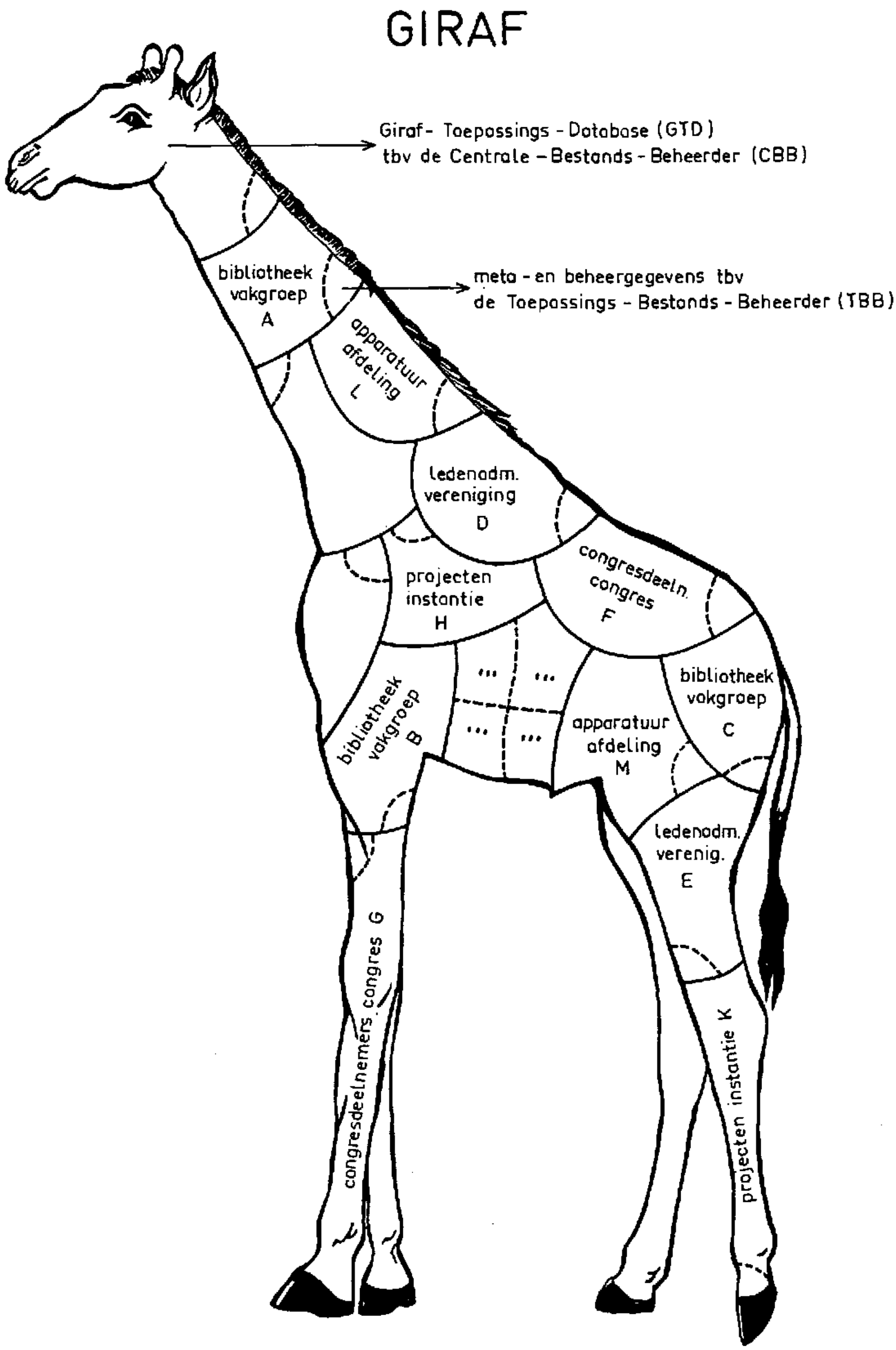
ment kan de TBB zijn GTD beheren. Dit betekent o.a., dat hij wachtwoorden op drie toegangsniveaus kan vaststellen. Het is de bedoeling, dat uitsluitend de TBB toegang heeft via toegangsniveau 1 (bijv. het wachtwoord 'ALLEENIK'). Geautoriseerde medewerkers hebben toegang via toegangsniveau 2 (bijv. het wachtwoord 'OOKANDER'). Toegangsniveau 3 kan worden vrijgegeven voor het raadplegen van de gegevens (bijv. het wachtwoord 'IEDEREEN'). Met behulp van functie 6, die alleen beschikbaar wordt gesteld, als een gebruiker via een wachtwoord van toegangsniveau 1 toelating tot de GTD heeft verkregen, kan de TBB nog meer maatregelen treffen. In 4.4.7 wordt hierop nader ingegaan.
Door systeemkenmerk en wachtwoorden niet vrij te geven kan een gebruiker geïsoleerd werken. Samenwerking tussen groeperingen is ook mogelijk. Zo kunnen bijvoorbeeld vakgroepbibliotheken samenwerken door wachtwoorden van toegangsniveau 3 uit te wisselen. De TBB houdt het heft in eigen hand, want hij kan op elk moment de wachtwoorden wijzigen. Op deze manier kan iedere TBB de muren om zijn GTD net zo hoog (of laag) maken, als hij dat wenst (zie figuur 7).
De overige functies zullen nu in het kort worden behandeld. Opgemerkt moet worden, dat functie 6 alleen beschikbaar is onder toegangsniveau 1 en de functies 4 en 5 alleen onder de toegangsniveaus 1 en 2. Het muteren in functie 2 en het verwijderen in functie 3 zijn eveneens alleen toegankelijk onder de toegangsniveaus 1 en 2.

4.4.2 *Helpfunctie (functie 1)*
De handleiding met betrekking tot het gebruik van 'GIRAF' is via deze functie beschikbaar. De gebruiker kan de gewenste hoofdstukken selecteren. TBB's, die over een eigen gebruikersafhankelijk subsysteem beschikken, kunnen een eigen hoofdstuk aan de handleiding toevoegen.

4.4.3 *Inbrengen van nieuwe referenties (functie 5)*
Het inbrengen van nieuwe referenties kan alleen gebeuren door de ToepassingsBestandsBeheerder (TBB) (toegangsniveau 1) of door geautoriseerde medewerkers (toegangsniveau 2). Er is veel aandacht besteed aan de gebruiksvriendelijkheid van deze functie. Allereerst kan de gebruiker kiezen tussen een serie schermbeelden voor zogenaamde standaardreferenties of een serie schermbeelden voor zeer uitgebreide referenties. De meeste gebruikers zullen kunnen werken met de serie schermbeelden voor standaardreferenties. Op alle schermbeelden verschijnen bij de gegevensgroepen de door de TBB gekozen namen. Op het standaardschermbeeld zijn 12 invoerregels beschikbaar, voor iedere gegevensgroep 3 regels. Dus bijvoorbeeld 3 regels voor namen, 3 regels voor de titel, 3 regels voor de locatie en 3 regels voor de abstract. Vanzelfsprekend kan er worden gewerkt met vervolgschermbeelden. Indien er echter wordt gewerkt met bijvoorbeeld een adressenbestand, is het niet zinvol om 3 regels te reserveren voor de naam. Het iedere keer overslaan van 2 regels wordt bovendien op den duur vervelend. Om dit probleem op te lossen kan een TBB via functie 6 (wijzigen bestandsgegevens) voor iedere gegevensgroep opgeven wat het gewenste aantal regels op het betreffende invoerschermbeeld is. Bovendien kan hij voor het invoerschermbeeld een initialiseringskarakter definiëren (bijvoorbeeld een '.' of een '-'). Indien dit initialiseringskarakter de letter 'H' ('H' van herhalen) is, zullen, nadat de gegevens in de database zijn opgenomen, de gegevens op de invoerschermbeelden blijven staan. Op die manier kunnen veel gelijksoortige gegevens snel worden ingevoerd. In literatuurbestanden wordt vaak gewerkt met een jaartal (+ suffix) van publikatie. Een TBB kan definiëren of hij wel of niet met een jaartal + suffix wil werken. Een TBB kan op elk moment (on-line) wijzigingen aanbrengen. Aangezien de invoerschermbeelden op uitvoeringstijd worden opgebouwd, hebben wijzigingen onmiddellijk effect. Met behulp van bovengenoemde parameters kan een TBB zijn 'eigen' invoerschermbeelden definiëren. Indien men dan ook tien willekeurige 'GIRAF'-gebruikers referenties ziet invoeren, kan men tien 'eigen' series invoerschermbeelden zien (zie figuur 8). Een andere mogelijkheid bij het invoeren van gegevens is het eerst kopiëren van de gegevens van een reeds bestaande referentie. Na aanpassing kunnen de gegevens van de nieuwe referentie in de GIRAF-Toepassings-Database (GTD) worden opgenomen.

Sommige gebruikers willen de gegevens, die tot een gegevensgroep behoren, via een vaste, specifieke indeling inbrengen. Een hulpmiddel daarbij wordt geboden door de mogelijkheid als referentie een dummy-referentie met een initialiseringsmasker te definiëren. Voordat men nu de nieuwe referentiegegevens inbrengt, kan de dummy-referentie met het masker worden gekopieerd op de invoerschermbeelden.

4.4.4 *Ophalen/muteren van een individuele referentie (functie 2)*
Het ophalen van de gegevens van een individuele referentie kan onder elk toegangsniveau plaatsvinden. Via functie 6 (wijzigen van bestandsgegevens) kan de ToepassingsBestandsBeheerder (TBB) vastleggen welke gegevensgroepen onder toegangsniveau 1, respectievelijk toegangsniveau 2 en toegangsniveau 3 beschikbaar worden gesteld. Het ligt voor de hand, dat met name onder toegangsniveau 3 niet alle gegevens van een referentie ter beschikking komen van de betreffende gebruiker. Dit is bijvoorbeeld het geval, als in de gegevensgroepen 1, 2 en 3 gegevens worden vastgelegd, die door veel gebruikers mogen worden geraadpleegd, terwijl gegevensgroep 4 gegevens bevat, die alleen voor een beperkte groep gebruikers toegankelijk zijn. Het *ophalen* van de gegevens van een individuele referentie kan plaatsvinden op basis van een *referentienaam* uit gegevensgroep 1 of het *referentienummer*. Een referentienummer is uniek en het resultaat is dan ook het beschikbaar komen van de referentiegegevens. Met een referentienaam zijn er wat meer mogelijkheden. Enerzijds kunnen aan een referentie verschillende namen toegekend zijn, bijvoorbeeld bij tijdschriftartikelen, anderzijds kan een naam aan verschillende referenties zijn verbonden, omdat een auteur verschillende tijdschriftartikelen heeft geschreven. In het laatste geval zal van alle referenties, waaraan de betreffende naam is verbonden de eerste regel van gegevensgroep 2 op het betreffende schermbeeld verschijnen (bijvoorbeeld de eerste regel van de titel), zodat de gebruiker de gewenste referentie kan kiezen. Wanneer als referentienaam een niet in de GIRAF-Toepassings-Database (GTD) aanwezige naam wordt opgegeven (of een *generieke naam*), wordt een schermbeeld geproduceerd met een lijst van in de GTD aanwezige referentienamen



Figuur 7: Structuur van de 'GIRAF'-database

die in de (alfabetisch geordende) omgeving van de ingevoerde naam liggen.
Het *wijzigen* van gegevens is alleen toegestaan onder de toegangsniveaus 1 en 2. De wijzigingen worden direct in de GTD aangebracht.

4.4.5 *Selecteren/verwijderen van referenties (functie 3)*
Het on-line selecteren van een verzameling referenties kan alleen plaatsvinden met behulp van het codesysteem. In 3 is reeds vermeld, dat het codesysteem bij 'GIRAF' bestaat uit een (in principe onbeperkt) aantal attribuutcodes en per attribuutcode een attribuutwaardenverzameling. Iedere gebruiker bepaalt zelf een aantal attribuutcodes en de daarbij behorende attribuutwaardenverzamelingen.
Bij het invoeren van een referentie kunnen voor de betreffende attribuutcodes één of meer attribuutwaarden aan de referentie worden toegekend (zie figuren 3 en 4). Voor een literatuurreferentie kunnen bijvoorbeeld worden gekozen attribuutcode 'CORE' (afkorting van COmputer REviews: een literatuurclassificatiemethode) met bijbehorende attribuutwaarden 'G.2.2' en 'G.3.3' en attribuutcode 'KAST' met bijbehorende attribuutwaarde 'KLAAS'. Een en ander betekent, dat de betreffende literatuurreferentie handelt over 'physical design database management' en 'information search and retrieval' en dat deze te vinden is in de kast bij medewerker 'KLAAS'. De gebruiker kan nu selecties maken door het opgeven van zogenaamde *items* en *relaties* tussen de items. Een item bestaat in dit geval uit een attribuutcode met één of meer attribuutwaarden. De relaties zijn EN- en OF-relaties.

'GIRAF' biedt de mogelijkheid om selectiecriteria d.m.v. *selectieprofielen* in de GIRAF-Toepassings-Database (GTD) op te slaan (zie 4.4.7). Door het intoetsen van het gewenste profielnummer kan het selectieprofiel op het beeldscherm verschijnen. Na het eventueel muteren van de criteria kan de selectie worden uitgevoerd.
'BIBINFO' bood de mogelijkheid om te selecteren met behulp van *trefwoorden*. Ook 'GIRAF' kent die mogelijkheid. Een ToepassingsBestandsBeheerder (TBB) kan een verzameling trefwoorden definiëren. Bij ieder trefwoord behoort een attribuutcode + attribuutwaarde. Op deze manier kunnen gebruikers selecties maken met trefwoorden en EN- en OF-relaties tussen trefwoorden.
Nadat de selectie heeft plaatsgevonden, kan de gebruiker kiezen, wat hij met de geselecteerde referenties wil doen. Een van de mogelijkheden is het bekijken van (een deel van) de gegevens op het beeldscherm. Een andere mogelijkheid is het afdrukken van de referenties op een door de gebruiker op te geven printer. Het aantal af te drukken referenties kan door de TBB worden beperkt (zie 4.4.7). De verzameling geselecteerde referenties kan ook op een achtergrondgeheugen worden geplaatst. Op die manier kunnen gebruikers een *extract* uit de database ter beschikking krijgen. Met behulp van de 'GIRAF'-rapportagemogelijkheden kunnen dan de diverse overzichten worden vervaardigd. Interessant is echter ook de mogelijkheid om met behulp van door de gebruiker zelf ontwikkelde programmatuur de extractgegevens te bewerken. Opgemerkt moet worden dat gebruikers in alle bovenstaande gevallen alleen die gegevens ter beschikking krijgen, waartoe zij volgens hun toegangsniveau toegang hebben. Het verwijderen uit de database van de verzameling geselecteerde referenties behoort eveneens tot de mogelijkheden.

4.4.6 *Opstarten batch-verwerking (functie 4)*
Functie 4 stelt de gebruiker in staat om, als onderdeel van on-line activiteiten, sommige standaard batch-tools op te starten. Met behulp van deze batch-tools kunnen diverse overzichten worden vervaardigd zoals de gebruikshandleiding, gebruiksstatistieken en bestandsstatistieken.

4.4.7 *Wijzigen bestandsgegevens (metagegevens) (functie 6)*
De TBB heeft een functie ter beschikking, waarmee hij zijn GTD kan beheren en *metagegevens* kan wijzigen. De volgende subfuncties kunnen worden onderkend:

4.4.7.1 *Wachtwoorden en autorisatiekenmerken*
Voor de toegangsniveaus 1, 2 en 3 zijn wachtwoorden en autorisatiekenmerken vastgelegd. Een van de autorisatiekenmerken betreft het *retrievalniveau*. De ToepassingsBestandsBeheerder (TBB) kan bijvoorbeeld bij toegangsniveau 3 opgeven, dat alleen de gegevens van de gegevensgroepen 1, 2 en 3 toegankelijk zijn. Bij de toegangsniveaus 1 en 2 kan hij dan alle gegevens beschikbaar stellen. Er zijn in 'GIRAF' een aantal combinaties van gegevensgroepen en attribuutgegevens gedefinieerd met een daarbij behorend nummer. De TBB kan per toegangsniveau het gewenste combinatienummer opgeven. Het opstarten van *batch-programma's* (zie 4.4.6) kan alleen plaatsvinden als de TBB het betreffende autorisatiekenmerk op 'JA' heeft gezet. Ook het *wijzigen van referenties* of het afdrukken van geselecteerde referenties kan alleen plaatsvinden, als de betreffende autorisatiekenmerken op 'JA' staan. De TBB kan bovendien per toegangsniveau een *limiet* stellen aan het aantal af te drukken referenties. Soms is het gewenst om de toegang tot een GIRAF-Toepassings-Database (GTD) alleen toe te staan vanaf één of meer vaste terminals. De TBB kan een dergelijke *terminalidentificatiebescherming* realiseren door per toegangsniveau de namen van de betreffende terminals vast te leggen. Op deze manier kan hij bijvoorbeeld toegangsniveau 1 alleen toestaan op de terminal, die bij de TBB op de kamer staat. Toegangsniveau 3 kan bijvoorbeeld geheel worden vrijgegeven of kan worden beperkt tot een aantal opgegeven terminals.

4.4.7.2 *Bestandsidentificatie en -informatie*
'GIRAF' bevat zo'n 75 schermbeelden (zie figuren 6 en 8). Alle schermbeelden hebben eenzelfde indeling. Zo bevat regel 1 o.a. altijd de naam van de instelling waar het systeem is geïmplementeerd.
Bij de TH-Delft staat vermeld: *** TECHNISCHE HOGESCHOOL DELFT ***. Regel 2 bevat o.a. de tekst 'GIRAF' gevolgd door de naam van de GTD waarmee wordt gewerkt. De TBB kan deze *bestandsidentificatie* vastleggen en naar believen wijzigen. Op die manier is op ieder schermbeeld te zien bij welke instelling en met welke GTD de gebruiker werkt. Naast de bestandsidentificatie kan de TBB *bestandsinformatie* vastleggen. Deze bestandsinformatie verschijnt tijdens de toegangsprocedure op het scherm. De TBB kan op die manier een korte beschrijving vastleggen om gebruikers op bepaalde zaken te attenderen. De TBB kan bovendien het *lettertype* definiëren. Dit houdt in, dat indien de TBB wenst te wer-

```
*****************************************************************************************
* *** TECHNISCHE HOGESCHOOL DELFT ***                                                   *
* *** GIRAF    : "Bestands-ident" ***   FUNKTIE : 5.1                                   *
*                                                                                       *
* *** INBRENGEN NIEUWE REFERENTIE ***   TE COPIEREN REFERENTIENUMMER : ........         *
*                                                                                       *
* NAAM     : ..............................                                             *
*          : ..............................                                             *
*          : ..............................          JAAR : .... ..                     *
* TITEL    : ...................................................................... *
*          : ...................................................................... *
*          : ...................................................................... *
* LOKATIE  : ...................................................................... *
*          : ...................................................................... *
*          : ...................................................................... *
* ABSTRACT : ...................................................................... *
*          : ...................................................................... *
*          : ...................................................................... *
*            ....+....1....+....2....+....3....+....4....+....5....+....6....+...*
*                                                                                       *
*          *1*  VERVOLG                        *3*  ATTRIBUTEN                          *
*          *2*  BEGIN REFERENTIE-GEGEVENS      *4*  COPIEREN REFERENTIENUMMER           *
*                                                                                       *
*              VUL DE GEWENSTE GEGEVENS IN EN MAAK UW OPTIE-KEUZE : _                   *
* -.-.-.-.-.-.-.-.-.-.-.-.-.-.-"systeem-boodschappen"-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.*
*****************************************************************************************
*****************************************************************************************
* *** TECHNISCHE HOGESCHOOL DELFT ***                                                   *
* *** GIRAF    : "Bestands-ident" ***   FUNKTIE : 5.1                                   *
*                                                                                       *
* *** INBRENGEN NIEUWE REFERENTIE ***   TE COPIEREN REFERENTIENUMMER : ........         *
*                                                                                       *
* NAAM     :                                                                            *
*          :                                                                            *
* OMSCHR   :                                                                            *
*          :                                                                            *
* SPECIFIC :                                                                            *
*          :                                                                            *
*          :                                                                            *
*          :                                                                            *
* DIVERSEN :                                                                            *
*          :                                                                            *
*          :                                                                            *
*          :                                                                            *
*            ....+....1....+....2....+....3....+....4....+....5....+....6....+...*
*                                                                                       *
*          *1*  VERVOLG                        *3*  ATTRIBUTEN                          *
*          *2*  BEGIN REFERENTIE-GEGEVENS      *4*  COPIEREN REFERENTIENUMMER           *
*                                                                                       *
*              VUL DE GEWENSTE GEGEVENS IN EN MAAK UW OPTIE-KEUZE : _                   *
* -.-.-.-.-.-.-.-.-.-.-.-.-.-.-"systeem-boodschappen"-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.*
*****************************************************************************************
*****************************************************************************************
* *** TECHNISCHE HOGESCHOOL DELFT ***                                                   *
* *** GIRAF    : "Bestands-ident" ***   FUNKTIE : 5.1                                   *
*                                                                                       *
* *** INBRENGEN NIEUWE REFERENTIE ***   TE COPIEREN REFERENTIENUMMER : ........         *
*                                                                                       *
* NAAM     : ______________________________                                             *
* ADRES    : ______________________________________________________________________*
*          : ______________________________________________________________________*
* WOONPL   : ______________________________________________________________________*
*          : ______________________________________________________________________*
* LEDENADM : ______________________________________________________________________*
*          : ______________________________________________________________________*
*          : ______________________________________________________________________*
*          : ______________________________________________________________________*
*          : ______________________________________________________________________*
*          : ______________________________________________________________________*
*          : ______________________________________________________________________*
*            ....+....1....+....2....+....3....+....4....+....5....+....6....+...*
*                                                                                       *
*          *1*  VERVOLG                        *3*  ATTRIBUTEN                          *
*          *2*  BEGIN REFERENTIE-GEGEVENS      *4*  COPIEREN REFERENTIENUMMER           *
*                                                                                       *
*              VUL DE GEWENSTE GEGEVENS IN EN MAAK UW OPTIE-KEUZE : _                   *
* -.-.-.-.-.-.-.-.-.-.-.-.-.-.-"systeem-boodschappen"-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.*
*****************************************************************************************
```

Figuur 8: Voorbeelden van invoerschermbeelden ken met uitsluitend hoofdletters of uitsluitend kleine letters, hij dit kan vastleggen. 'GIRAF' zorgt er dan voor, dat alvorens gegevens in de database worden opgeslagen alle letters worden omgezet in hoofdletters respectievelijk in kleine letters. De verstekwaarde is hoofd- en kleine letters. In het laatste geval worden de gegevens opgeslagen, zoals ze door de gebruiker zijn ingevoerd. In bijzondere gevallen kan het gewenst zijn, dat gegevens in een *versluierde* vorm op het achtergrondgeheugen worden geplaatst. De TBB kan dit echter niet zelf realiseren. Hij moet hiervoor een gemotiveerd verzoek indienen. Als op het verzoek positief wordt gereageerd, zal de CBB de betreffende optie activeren. Een en ander is op die manier geregeld, omdat het versluieren van gegevens nogal wat computertijd vraagt. Het versluieren van gegevens zal dan ook worden beperkt tot de werkelijk noodzakelijke gevallen.

4.4.7.3 *Systeemboodschappen*
Niet alleen de regels 1 en 2 van alle schermbeelden hebben een vaste indeling. Ook regel 24 heeft een vaste indeling. Deze regel is namelijk gereserveerd voor systeemboodschappen. De TBB kan *systeemboodschappen* versturen. Hij heeft op die manier als het ware een 'intercom' mogelijkheid tot zijn beschikking. De TBB kan per functie een systeemboodschap vastleggen. Dit betekent dat ieder schermbeeld van een functie de betreffende systeemboodschap zal bevatten. Pas nadat de TBB de betreffende systeemboodschap heeft verwijderd zal deze niet meer verschijnen. Naast de systeemboodschappen per functie heeft de TBB de mogelijkheid om een algemene systeemboodschap te definiëren. Deze heeft altijd voorrang boven de functiegerichte systeemboodschappen. Een algemene systeemboodschap '*** ER IS KOFFIE MET GEBAK BIJ ... ***' zal op alle schermbeelden van de betreffende GTD verschijnen en vermoedelijk leiden tot grote drukte bij de betreffende persoon.

4.4.7.4 *Presentatie van de gegevensgroepen*
In 4.4.3 (inbrengen nieuwe referenties) is reeds ingegaan op de mogelijkheden voor de ToepassingsBestandsBeheerder (TBB) om per gegevensgroep een eigen naam vast te leggen. Ook het vastleggen van een initialiseringsteken voor de invoerschermbeelden en het aantal regels per gegevensgroep alsmede het al dan niet gebruiken van een jaartal + suffix bij het invoeren van referenties is daar reeds aan de orde geweest.

4.4.7.5 *Attribuutwaardeomschrijvingen, trefwoorden*
De TBB heeft de mogelijkheid om aan een attribuutcode + attribuutwaarde een omschrijving en één of meer trefwoorden te koppelen. De *attribuutwaardeomschrijving* speelt een rol bij functie 2, het ophalen van individuele referenties. Hoewel het codesysteem, met alfanumerieke attribuutcodes van 4 posities en alfanumerieke attribuutwaarden van 6 posities, reeds mogelijkheden biedt om suggestieve codes en waarden te kiezen, is het soms prettig om bij het opvragen van een referentie van een betreffende attribuutcode + attribuutwaarde een ruimere omschrijving op het beeldscherm te krijgen. De TBB heeft een mogelijkheid om deze attribuutwaardeomschrijving te koppelen aan een attribuutcode + attribuutwaarde. Een andere rol spelen de *trefwoorden*. Bij het selecteren van referenties willen sommige gebruikers graag werken met trefwoorden (zie 4.4.5). De TBB kan één of meer trefwoorden (synoniemen) koppelen aan een attribuutcode + attribuutwaarde. Gebruikers kunnen nu selecties maken met behulp van de trefwoorden. 'GIRAF' zal intern de trefwoorden omzetten in combinaties van attribuutcode + attribuutwaarde.

4.4.7.6 *Selectieprofiel, uitvoerlayout, batch-verstekwaarden*
Met behulp van *selectieprofielen* kan een ToepassingsBestandsBeheerder (TBB) selectiecriteria vastleggen in de GIRAF-Toepassings-Database (GTD). Dit is vooral van belang bij regelmatig terugkerende selecties. Ook kan de TBB op die manier een aantal selecties als het ware klaarzetten voor gebruikers van zijn GTD. Naast selectieprofielen kan een TBB ook *profielen voor uitvoerlayouts* in de GTD vastleggen. Ten behoeve van het opstarten van batch-verwerking door gebruikers kan de TBB een aantal *verstekparameterwaarden* en andere gegevens, o.a. met betrekking tot de Job Control Language, vastleggen. De TBB kan bovendien de on-line beschikbare *gebruikershandleiding* uitbreiden met zijn 'eigen' hoofdstuk. Met behulp van deze mogelijkheden kan een TBB een gebruiksvriendelijk klimaat scheppen rondom zijn GTD.

4.4.8 *Gebruikersafhankelijke on-line functie (functie 7)*
Het gebruikersafhankelijke subsysteem (zie 4.3) biedt o.a. de mogelijkheid om voor een GTD een 'eigen' online functie te maken. Zo heeft de Centrale BestandsBeheerder (CBB) een eigen on-line functie. Sommige gebruikersafhankelijke on-line functies zullen worden gemaakt voor bepaalde *groepen* GTD's. Daarbij kan bijvoorbeeld gedacht worden aan een eigen on-line functie ten behoeve van vakgroepbibliotheken en apparatuurbeheerders, die behoefte hebben aan een eenvoudige uitleenadministratie. Een andere, voor de hand liggende, toepassing zou een eenvoudige ledenadministratie zijn ten behoeve van bijvoorbeeld het Muzisch Centrum en andere groeperingen, die al met de oude systemen hun ledenadministratie hebben geautomatiseerd.

5 GEBRUIKSVRIENDELIJKHEID

De opdracht was om een gebruiksvriendelijk systeem te bouwen. In de diverse paragrafen is reeds aandacht besteed aan de gebruiksvriendelijkheid van 'GIRAF'. Samenvattend kunnen de volgende punten onder de term gebruiksvriendelijk worden gerangschikt:

a. Het *opstarten* van 'GIRAF' door een gebruiker is eenvoudig. Door het woord 'GIRAF' in te toetsen wordt hij geconfronteerd met de toegangsschermbeelden van 'GIRAF'. Gebruikers merken er in principe niets van dat 'GIRAF' werkt met het draagsysteem IDMS DB/DC.
b. De gebruiker kan bij de *invoerfunctie* (zie 4.4.3) kiezen of hij wil werken met een serie invoerschermbeelden voor standaardreferenties of met een serie invoerschermbeelden voor uitgebreide referenties. Iedere ToepassingsBestandsBeheerder (TBB) kan zijn eigen naamgeving kiezen voor de gegevensgroepen. Tevens kan hij het aantal regels vastleggen voor de respectievelijke gegevensgroepen op de invoerschermbeelden. Door daarnaast een initialiseringsteken op te geven en te vermelden

of hij al dan niet wil werken met een jaartal (+ suffix) van publikatie heeft de TBB voldoende mogelijkheden om de invoerschermbeelden een eigen gezicht te geven en af te stemmen op zijn eigen werkomgeving. Het invoeren van referenties wordt vergemakkelijkt door het kunnen werken met een invoermasker en de mogelijkheid om een bestaande referentie te kopiëren en na mutatie in de database op te nemen.

c. Alle 75 schermbeelden hebben een systematische indeling. De regels 1, 2 en 24 hebben een *uniforme indeling*, terwijl bij de indeling van de regels 3 t/m 23 is gestreefd naar een systematische indeling. Zo heeft bijvoorbeeld de optie 'VORIG NIVEAU' op alle schermbeelden, waar deze optie voorkomt, hetzelfde nummer. Dit voorkomt vergissingen. Met deze optie kan een gebruiker teruggaan naar een vorig schermbeeld. Dit is bijvoorbeeld interessant, als hij een te grove selectie heeft gemaakt en alsnog een verfijning wil aanbrengen. De reeds ingebrachte selectiecriteria blijven bij de 'vorige schermbeelden' bewaard en hoeven dus niet opnieuw te worden ingetoetst.

d. Gebruikers van 'GIRAF' kunnen doorgaans van het ene schermbeeld naar het andere schermbeeld komen door het intoetsen van een optienummer. Ervaren gebruikers kennen veel schermbeelden en willen daarom vaak één of meer schermbeelden van een serie schermbeelden overslaan. 'GIRAF' bevat op een aantal punten de mogelijkheid van *'command chaining'*, waarmee één en ander bewerkstelligd kan worden.

e. Het *op maat* maken van 'GIRAF' door het vervangen van dialogen of door toevoeging van een gebruikersafhankelijk subsysteem is reeds aan de orde geweest in 4.3 en 4.4.8.

f. Een gebruiker kan desgewenst geïsoleerd werken ('eenzaam maar niet alleen') of kan *samenwerken* met anderen binnen de TH-Delft of daarbuiten (zie figuur 7). Hij bepaalt grotendeels zelf hoever die samenwerking gaat.

g. Door te werken met *selectieprofielen*, profielen voor uitvoerlayouts en verstekwaarden voor batchverwerking kan een TBB het zichzelf en zijn gebruikers gemakkelijk maken.

h. Het op een later tijdstip implementeren van andere *taalversies* van 'GIRAF' is eenvoudig te realiseren. Een Engelstalige versie zal op korte termijn beschikbaar komen.

i. Een *gebruikshandleiding* is zowel op papier als on-line beschikbaar. Iedere TBB kan desgewenst zijn eigen hoofdstuk toevoegen. Naast deze gebruikshandleiding is er nog een verkorte gebruikshandleiding. Deze laatste is bedoeld voor gebruikers die, zonder zich uitgebreid te willen verdiepen in alle mogelijkheden van 'GIRAF', snel met hun (eenvoudige) toepassing van start willen gaan.

6 PRIVACYASPECTEN

Binnen 'GIRAF' is op een aantal punten aandacht besteed aan privacyaspecten:

a. De on-line toegang tot een GIRAF-Toepassings-Database (GTD) wordt geregeld met behulp van *systeemkenmerken* en *wachtwoorden*. De systeemkenmerken en wachtwoorden staan versluierd op het achtergrondgeheugen. Per toegangsniveau kan een ToepassingsBestandsBeheerder (TBB) de toegestane handelingen vastleggen en definiëren tot welke gegevens toegang wordt verleend. Bovendien kan een TBB met behulp van *terminalidentificatiebescherming* toegang tot zijn GTD koppelen aan bepaalde terminals. Zelfs kan hij deze terminals fysiek beveiligen door ze van een *slot* te laten voorzien.

b. Toegang tot een GTD via batch-verwerking is beveiligd met behulp van wachtwoorden.

c. Printuitvoer kan via een *lokale printer* worden afgeleverd.

d. Het *versluieren* van gegevens op het achtergrondgeheugen behoort tot de mogelijkheden.

e. Een TBB kan desgewenst een *mutatieverslag* krijgen. Hierop staan alle mutaties vermeld, die op zijn GTD zijn aangebracht. Per mutatie worden o.a. vermeld: datum, tijdstip, wachtwoord en terminal via welke de mutatie is ingevoerd. Op die manier kan een TBB, weliswaar achteraf, nagaan wat er met zijn GTD is gebeurd.

f. De *CBB* heeft toegang tot enkele metagegevens van de GTD's. Zo is de CBB bijvoorbeeld bekend met het systeemkenmerk van een GTD, omdat hij dit systeemkenmerk zelf aanbrengt. De CBB heeft echter zonder toestemming van een TBB geen toegang tot diens GTD.

Er kan worden geconcludeerd, dat binnen 'GIRAF' veel aandacht is besteed om de privacy van gegevens in een GTD te waarborgen. De genomen maatregelen zijn echter niet voldoende. De *omgeving* waarbinnen 'GIRAF' gaat functioneren, is mede bepalend voor het effect van de getroffen maatregelen. Immers, een gevangenis met enkele dikke muren, maar met een open tuinhekje aan de achterkant biedt de maatschappij niet die bescherming, die zij verwacht.

7 BEVEILIGINGSASPECTEN

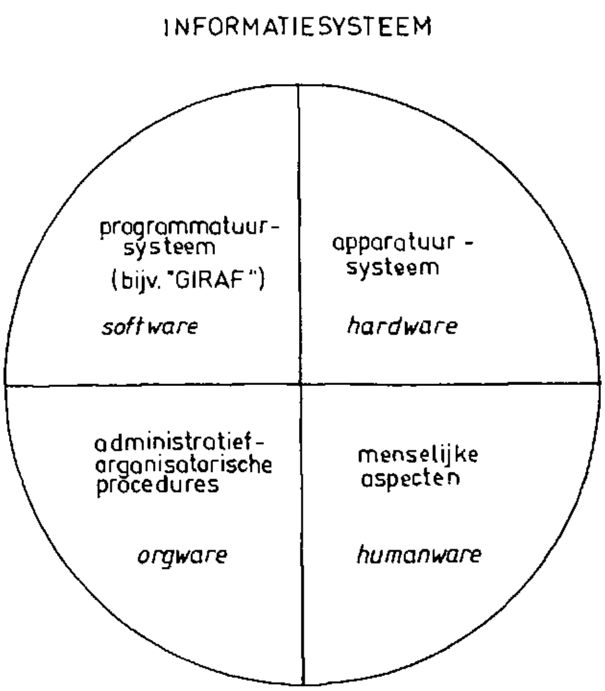
'GIRAF' is ontwikkeld met behulp van het database-management- en datacommunicatie-systeem IDMS DB/DC. Dit draagsysteem heeft mogelijkheden om maatregelen te treffen tegen het verloren gaan van gegevens zoals recovery/restart, journal files, logging files, checkpointing, roll back, etc. Het Rekencentrum van de TH-Delft biedt mogelijkheden om GTD's te kopiëren en elders op te slaan. In het kader van dit artikel gaat het te ver om op deze beveiligingsaspecten nader in te gaan. Er kan worden geconcludeerd dat er voldoende mogelijkheden zijn om de gegevensbestanden te beschermen tegen calamiteiten. Wel is het voor sommige **informatiesystemen** (zie figuur 9) noodzakelijk dat er een noodprocedure beschikbaar is, zodat bij het uitvallen van de computer het informatiesysteem kan blijven functioneren.

8 TOEPASSINGEN

De bestanden van enkele tientallen gebruikers van de oude systemen 'BIBLIOSYSTEM' en 'BIBINFO' zijn inmiddels geconverteerd of worden binnenkort geconverteerd naar 'GIRAF'. Enkele daarvan zijn:

a. Diverse GIRAF-Toepassings-Databases (GTD's) die literatuurreferenties bevatten.

b. Diverse GTD's die de gegevens van een boekenverzameling bevatten.
c. Een GTD die de ledenadministratie van het Muzisch Centrum bevat.
d. Een GTD die de apparatuurgegevens bevat van de afdeling Scheikunde.
e. Het Transferpunt van de TH-Delft functioneert als intermediair tussen de TH en het bedrijfsleven. Het heeft tot doel de overdracht van kennis en deskundigheid vanuit de TH naar met name kleine en middelgrote ondernemingen te bevorderen. Het Transferpunt heeft enkele GTD's op het gebied van: adressen, beschikbare kennis binnen de TH-Delft, aanvragen tot kennisoverdracht, projecten, etc.
f. Een GTD die de gegevens van alle correspondentie van de afdeling ISO van het Rekencentrum bevat. De secretaresse van de afdeling ISO kan daarmee o.a. de medewerkers van de afdeling maandelijks een rappellijst sturen van de correspondentie die zij nog niet afgehandeld hebben.



Figuur 9: Componenten van een informatiesysteem

Met de ingebruikneming van het **programmatuursysteem** (zie figuur 9) 'GIRAF' is het Rekencentrum van de TH-Delft in staat om de komende jaren bepaalde **klassen van informatiesystemen** op zeer snelle wijze te ontwikkelen.

9 LITERATUUR

1. Application Development System/Online. Summary Description. Cullinet Software, 1983
2. Application Development System/Online. Application Design Guide, release 1.1. Cullinet Software, augustus 1983
3. Dijk, A. J. van; 'BIBLIOSYSTEM'. Computerhulp bij het samenstellen van bibliografieën. In: *Informatie*, jaargang 13 nr. 10 (oktober 1971), pp. 448-455
4. Dijk, A. J. van: 'BIBINFO' - Een real-time retrieval systeem. Afstudeerverslag, vakgroep Informatica TH-Delft. Memorandum no. RC-DMS-76 002, november 1976, 236 pag.
5. Dijk, A. J. van; Ervaringen met het COMMAND-level van CICS/VS bij de ontwikkeling van een geautomatiseerde bibliotheek-toepassing. In; *Informatie*, jaargang 22 nr. 4, april 1980, pp. 313-322
6. Dijk, A. J. van en R. de Roos; 'GIRAF' - General Information RetrievAl Facilities Functioneel ontwerp. Technische Hogeschool Delft, Rekencentrum Memorandum no. RC-ISO-82 012, december 1982

Publication 10

ADS/ONLINE

Ervaringen met een "ontwikkeltool" bij de ontwikkeling van het

information retrieval systeem GIRAF

Experiences with the development tool ADS/ONLINE
at the development of a computerised
information retrieval system

1984

Quality is debatable.

Aart van Dijk

This article previously appeared in:

Informatie, November 1984, pages 912- 927

Abstract

At the Dutch University of Technology (DUT) the 'GIRAF' information retrieval system was developed. An article on the functional aspects of this software system was published in the Dutcb magazine Informatie of May 1984. At development of 'GIRAF', the IDMS DB/DC database management and data communication system acted as a support system. Furthermore, the development tool 'Application Development System (ADS/ONLINE)' was used.
In this article, some of the technical aspects that played a part in the development of 'GIRAF' come up for discussion. It particularly describes the experiences with ADS/ONLINE in relationship with display image communication/dialogue communication, multiple databases and prototyping.

ADS/ONLINE

ERVARINGEN MET EEN 'ONTWIKKELTOOL' BIJ DE ONTWIKKELING VAN HET INFORMATION RETRIEVAL SYSTEEM GIRAF

door ir. A. J. van Dijk

Bij de Technische Hogeschool Delft is het information retrieval systeem 'GIRAF' ontwikkeld. Over de functionele aspecten van dit programmatuursysteem is gepubliceerd in Informatie, *mei 1984.*
Bij de ontwikkeling van 'GIRAF' heeft het database-management- en datacommunicatiesysteem IDMS DB/DC als draagsysteem gefungeerd. Bovendien werd gebruik gemaakt van het ontwikkeltool 'Application Development System (ADS/ONLINE)', dat evenals IDMS DB/DC, door de firma Cullinet Software geleverd wordt.
In dit artikel komen enkele technische aspecten, die bij de ontwikkeling van 'GIRAF' een rol hebben gespeeld, aan de orde. Met name wordt ingegaan op de ervaringen die zijn opgedaan met ADS/ONLINE in relatie tot schermbeeldcommunicatie/dialoogcommunicatie, multiple databases en prototyping.

1 INLEIDING

Het vervaardigen van een informatiesysteem is geen eenvoudige zaak. Er zijn diverse oorzaken die het ontwikkelen en invoeren van informatiesystemen doen mislukken [21]. Gebruikers van, met succes ontwikkelde en ingevoerde, informatiesystemen zorgen bovendien, doorgaans binnen korte tijd na oplevering van het informatiesysteem, voor wensen met betrekking tot uitbreidingen en/of aanpassingen van het informatiesysteem. Ooninex [21] rekent de mogelijkheden tot uitbreidingen en aanpassingen van een informatiesysteem tot de flexibiliteitsaspecten van een informatiesysteem:
'Tot het aspekt der flexibiliteit moet ook gerekend worden de mogelijkheid het ontworpen informatiesysteem naderhand nog uit te breiden met nieuwe informatiesubsystemen die bij het ontwerp niet voorzien waren ('open-ended'-informatiesysteem). Als het informatiesysteem is opgezet kan het ook onder omstandigheden werken die verschillen met die ten tijde van de opzet ervan. Dit eist aanpassingsvermogen ('adaptability')'.
In het meinummer 1984 van *Informatie* [14] is een artikel gepubliceerd dat de functionele aspecten van het programmatuursysteem 'GIRAF' beschrijft. 'GIRAF' (General Information RetrievAl Facilities) is een programmatuursysteem dat met behulp van de System Development Methodology (SDM) [17] is ontwikkeld door de afdeling Ontwikkeling Informatiesystemen (ISO) van het Rekencentrum van de Technische Hogeschool Delft (kortweg TH-Delft). Met behulp van 'GIRAF' is het Rekencentrum van de TH-Delft in staat om bepaalde klassen van informatiesystemen op zeer snelle wijze te ontwikkelen. Craenen [9] hanteert het begrip 'infosystemen' voor algemeen toepasbare programmatuur-/informatiesystemen voor eindgebruikers. 'GIRAF' kan tot de categorie der 'infosystemen' worden gerekend. In hoofdstuk 2 van dit artikel wordt een korte uiteenzetting van 'GIRAF' gegeven.
Bij de technische bouw van 'GIRAF' is o.a. gebruik gemaakt van een Amdahl 470V/7B computer en van het 'ontwikkeltool' **Application Development System/Online** (kortweg ADS/ONLINE) [1,2,3,4] van de firma Cullinet Software. Dit ontwikkeltool wordt ook gebruikt bij de ontwikkeling van een geautomatiseerd magazijnadministratie- en voorraadbeheersysteem ('GEMS') [16]. Leveranciers van programmatuur doen regelmatig aankondigingen van binnenkort te verschijnen nieuwe versies van hun produkten. Het ontwikkelen van een programmatuursysteem kan echter alleen plaatsvinden met de voor de gebruiker (systeemontwerper) beschikbare versies. De technische beslissingen inzake 'GIRAF' zijn genomen in **december 1983** op basis van de op **dat moment beschikbare** softwareprodukten. Eventuele latere aankondigingen van nieuwe versies zijn niet meer in de beschouwingen betrokken. In hoofdstuk 3 wordt een aantal basisbegrippen met betrekking tot ADS/ONLINE behandeld zoals het begrip 'dialoog', de samenhang tussen dialogen, het vervaardigen van schermbeelden en de relatie van ADS/ONLINE tot (pseudo-)conversationele programmering. De ervaringen, op systeemontwerpniveau, die zijn opgedaan met ADS/ONLINE bij de ontwikkeling van de systemen 'GIRAF' en 'GEMS' komen aan de orde in hoofdstuk 4. Ervaringen met betrekking tot responsietijden, geheugenbeslag, etc. komen niet aan de orde. Wellicht kunnen andere ADS/ONLINE gebruikers deze aspecten in een volgend artikel belichten. Paragraaf 4.2 handelt over de communicatie tussen dialogen. Daarbij wordt aangegeven op welke wijze aan de flexibiliteitsaspecten **open-ended** en **adaptability** gestalte is gegeven. Andere onderwerpen die aan bod komen zijn **prototyping** en het werken met **multiple databases**.
Voor lezers die slechts in enkele van de in dit artikel behandelde onderwerpen geïnteresseerd zijn volgt hier de inhoudsopgave:

1 INLEIDING
2 DE TOEPASSING
3 ADS/ONLINE
3.1 Inleiding
3.2 Dialogen
3.3 Samenhang tussen dialogen

3.4 OnLine Mapping (OLM)
3.5 ADS/ONLINE versus (pseudo-)conversationele programmering
4 ADS/ONLINE IN RELATIE TOT DE TOEPASSING
4.1 Inleiding
4.2 Communicatie tussen dialogen
4.3 Multiple databases
4.4 Prototyping
5 INDICES
6 CONCLUSIES
7 LITERATUUR/REFERENTIES

2 DE TOEPASSING

2.1 Inleiding

In 1970 is bij de TH-Delft het systeem 'BIBLIOSYSTEM' [11] ontwikkeld. Aanleiding voor de bouw van dit systeem was de grote hoeveelheid tijd, die onderzoekers moesten besteden aan het verzamelen, registreren en ordenen van literatuur. De klassieke wijze van rapporteren van een literatuurstudie was zeer tijdrovend o.a. vanwege de omvang van het typewerk. Bibliografieën waren soms op het moment dat ze gereed waren voor verspreiding alweer verouderd. Een en ander leidde tot het idee om het samenstellen van bibliografieën te automatiseren. 'BIBLIOSYSTEM' werd begin 1971 door de TH-Delft in gebruik genomen. In de loop der jaren bleek, dat door de 'abstracte' opzet het systeem bruikbaar was voor allerhande toepassingen. Naast bestanden met literatuurreferenties werden met behulp van 'BIBLIOSYSTEM' adresbestanden, bestanden voor ledenadministraties, apparatuurbestanden, diabestanden, boekbestanden, etc. vervaardigd, bijgewerkt en afgedrukt.

In 1976 ontwikkelde het Rekencentrum op verzoek van een congresorganisatiecomité het real-time systeem 'BIBINFO' [12]. Dit systeem werd als het ware bovenop 'BIBLIOSYSTEM' gebouwd. Het opbouwen van de bestanden gebeurde met 'BIBLIOSYSTEM'. Het raadplegen en het on-line selecteren gebeurde met 'BIBINFO'. 'BIBLIOSYSTEM' (1970) en 'BIBINFO' (1976) zijn ontwikkeld voor een beperkte doelgroep met methoden en hulpmiddelen, die in die jaren modern waren. Het grote succes van deze systemen binnen de TH-Delft en de voortdurende vraag naar dergelijke faciliteiten heeft de directie van het Rekencentrum in 1982 doen besluiten een nieuw information retrieval systeem te laten ontwikkelen, dat op een ruimere doelgroep moest worden gericht. Het moest de systemen 'BIBLIOSYSTEM' en 'BIBINFO' vervangen en zeer gebruiksvriendelijk zijn. Dit nieuwe systeem is het programmatuursysteem 'GIRAF' geworden.

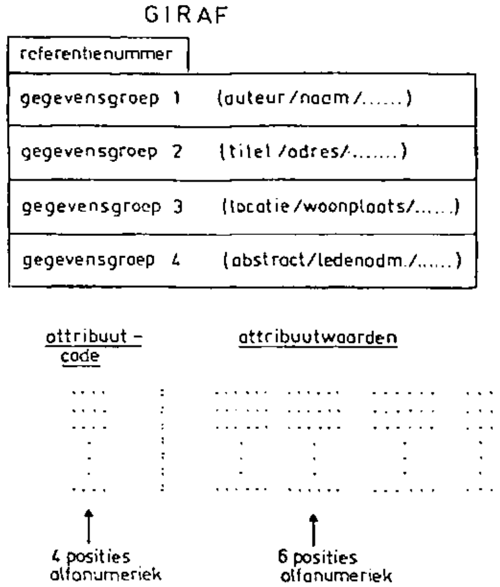
2.2 Uitgangspunten 'GIRAF'

Voordat met de bouw van 'GIRAF' is gestart is een aantal eisen geformuleerd waaraan het systeem moest voldoen [14]. De eisen kunnen als volgt worden samengevat:

- Uit het succes van de systemen 'BIBLIOSYSTEM' en 'BIBINFO' is de conclusie getrokken, dat de filosofie, waarop deze systemen waren gebaseerd, voor de TH-Delft een goede filosofie is. De eerste voorwaarde, die aan 'GIRAF' is gesteld, is het handhaven van deze filosofie. Dit betekende o.a., dat 'GIRAF' een **eenvoudig** en **doeltreffend** information retrieval systeem moest worden (eis 1).
- Aangezien 'GIRAF' voor een ruimere doelgroep moest worden gemaakt, moest het algemener bruikbaar zijn dan de oude systemen (eis 2).
- De gegevenseenheid in de oude systemen was de referentie. Toegang tot een **individuele** referentie geschiedde via het referentienummer of de auteur. Toegang tot een **verzameling** referenties was alleen mogelijk via een codesysteem. Dit codesysteem bestond uit een tiental attributen met voor ieder attribuut een waardenverzameling. Aan iedere referentie kon per attribuut één waarde uit de betreffende waardenverzameling worden toegekend. Binnen 'GIRAF' moest het codesysteem ruimer zijn (eis 3).
- De TH-Delft is een pluriforme organisatie, die veel zelfstandig opererende groeperingen herbergt. Sommige van die groeperingen willen (delen van) hun informatievoorziening op een geïsoleerde manier realiseren (bijv. promovendi) (eis 4). Andere groeperingen willen graag samenwerken met groeperingen binnen of buiten de TH-Delft. Voor die groeperingen is het noodzakelijk om vanuit verschillende locaties, met behulp van on-line-verwerking, bestanden te kunnen opbouwen, raadplegen en/of muteren (eis 5).
- 'GIRAF' moet kunnen worden toegepast ten behoeve van veel informatiesystemen binnen de TH-Delft. Daarom moet 'GIRAF' worden aangeboden als **confectiesysteem**. De opzet moest echter dusdanig zijn, dat het met weinig inspanning kan worden aangepast. Op die manier kan 'GIRAF' voor sommige gebruikers als **maatconfectiesysteem** fungeren (eis 6).
- Bij de TH-Delft werken en studeren ook buitenlanders. Het is daarom wenselijk dat 'GIRAF' kan werken met andere talen. Het systeem moest zodanig worden opgezet, dat in een later stadium **andere taalversies** snel kunnen worden geïmplementeerd (eis 7).
- 'GIRAF' moest voorzieningen hebben om de gebruikers van 'BIBLIOSYSTEM' en 'BIBINFO' op een zeer soepele wijze te kunnen laten overstappen naar het nieuwe systeem (eis 8).
- Door het Rekencentrum van de TH-Delft is als draagsysteem ten behoeve van database-management- en datacommunicatie-activiteiten gekozen voor IDMS DB/DC, dat wordt geleverd door de firma Cullinet Software. 'GIRAF' diende met behulp van IDMS DB/DC te worden ontwikkeld (eis 9).

2.3 Indeling van de referenties

De eenheid in de systemen 'BIBLIOSYSTEM' en 'BIBINFO' was de referentie. Hoewel oorspronkelijk bedoeld voor het behandelen van literatuurreferenties werden de systemen voor allerhande toepassingen gebruikt. De naamgeving was echter gebaseerd op literatuurreferenties. Zo was er sprake van een auteur, een titel, een locatie en een abstract. Daarnaast werd aan een referentie voor een aantal (maximaal 10) attributen een attribuutwaarde toegekend. Aangezien de term 'referentie' bij veel gebruikers van de oude systemen is ingeburgerd, is besloten deze term in 'GIRAF' over te nemen.
'GIRAF' moest een algemeen karakter hebben, daarom kon niet worden volstaan met de bestaande naamgeving.



Figuur 1: Gegevensmodel van een referentie binnen 'GIRAF'

Binnen 'GIRAF' bestaat een referentie uit vier gegevensgroepen (zie figuur 1). Iedere gebruiker die verantwoordelijk is voor een gegevensverzameling (in 'GIRAF'-termen: iedere ToepassingsBestandsBeheerder (TBB)) kan zelf de namen kiezen, die hij aan deze gegevensgroepen wenst te geven. Overal waar de gegevensgroepen worden gebruikt (op beeldscherm of papier) worden de door de TBB opgegeven namen gebruikt. 'GIRAF' kent zelf aan iedere nieuwe referentie een referentienummer toe.
Het codesysteem, waarmee attribuutcodes en attribuutwaarden aan een referentie kunnen worden toegekend, is uitgebreid (zie figuren 1 en 2). Het aantal attribuutcodes is nu onbeperkt. Bovendien kunnen per referentie aan een attribuutcode verschillende alfanumerieke attribuutwaarden worden toegekend.

2.4 Subsystemen

Het programmatuursysteem 'GIRAF' is opgedeeld in 4 subsystemen, te weten:

- subsysteem 1: conversie/opbouw database
- subsysteem 2: on-line activiteiten
- subsysteem 3: standaard batch-tools
- subsysteem 4: gebruikersafhankelijke toepassingen

Figuur 2: Voorbeeld van een 'GIRAF'-referentie

```
00000427
Roos,R. de
Speelgraagsingel 392
Hofje 3
2669 KL  Delft
bij  Den Haag
Betaling                 :  behoeft niet meer dan fl. 110,- te betalen
Telefonisch bereikbaar   :  079-896357
Organisatie              :  is in de toekomst beschikbaar voor
                            een bestuursfunctie
Bijzonderheden           :  spreekt chinees
```

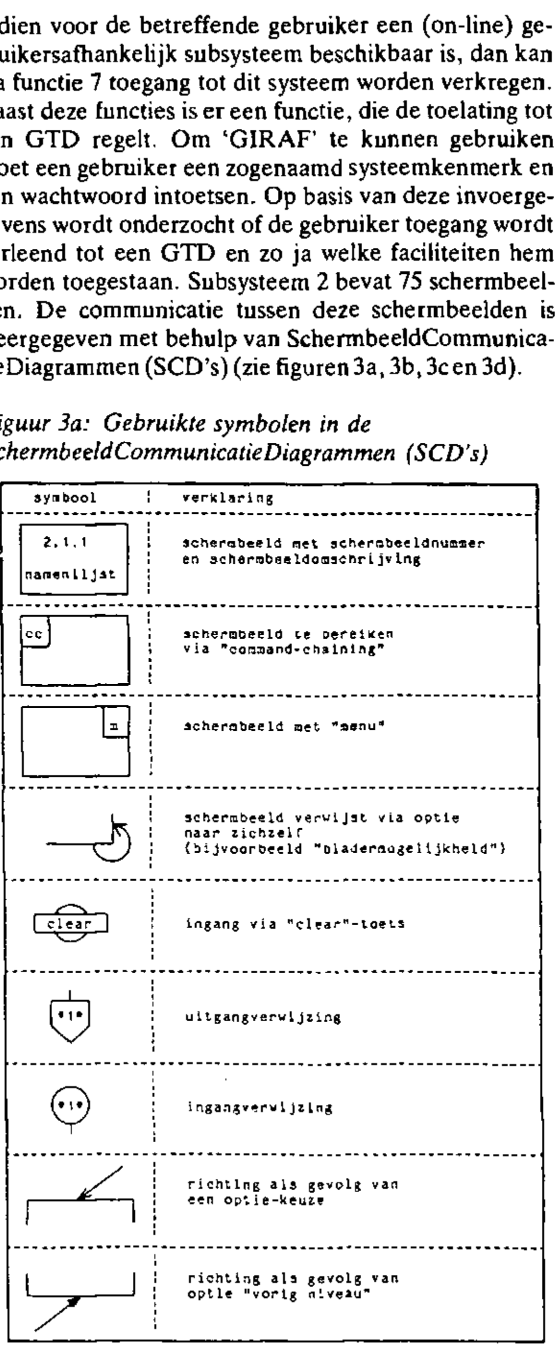
Attribuut-code	Attribuutwaarde	
BETA	BS-2	(betaling)
INST	PIANO	(bespeelt instrument)
INST	GITAAR	
INST	KLARNT	
CURS	DRUMS	(volgt cursus)
CURS	JAZZ	
TAAL	NLD	(spreekt taal)
TAAL	ENG	
TAAL	CHIN	
STAT	TM-MED	(status)
JAAR	1980	(jaar van 1e inschrijving)

Met behulp van **subsysteem 1** (conversie/opbouw database) kunnen bestanden in 'BIBLIOSYSTEM'-formaat worden geconverteerd naar een 'GIRAF-Toepassings-Database' (GTD).
Het on-line subsysteem (**subsysteem 2**) bestaat uit de volgende functies:

1 help (handleiding)
2 ophalen/muteren individuele referentie
3 selecteren/verwijderen verzameling referenties
4 opstarten serieverwerking (batch)
5 inbrengen nieuwe referentie(s)
6 wijzigen bestandsgegevens
7 *gebruikersafhankelijke functie*
9 afsluiten sessie

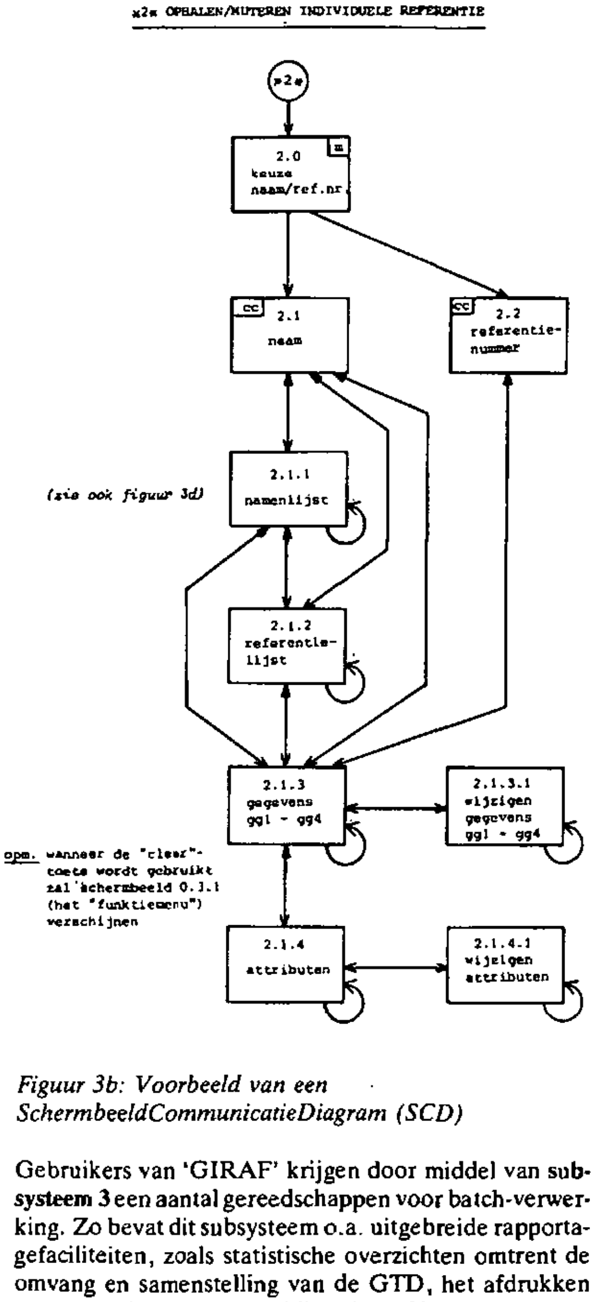
Indien voor de betreffende gebruiker een (on-line) gebruikersafhankelijk subsysteem beschikbaar is, dan kan via functie 7 toegang tot dit systeem worden verkregen. Naast deze functies is er een functie, die de toelating tot een GTD regelt. Om 'GIRAF' te kunnen gebruiken moet een gebruiker een zogenaamd systeemkenmerk en een wachtwoord intoetsen. Op basis van deze invoergegevens wordt onderzocht of de gebruiker toegang wordt verleend tot een GTD en zo ja welke faciliteiten hem worden toegestaan. Subsysteem 2 bevat 75 schermbeelden. De communicatie tussen deze schermbeelden is weergegeven met behulp van SchermbeeldCommunicatieDiagrammen (SCD's) (zie figuren 3a, 3b, 3c en 3d).

Figuur 3a: Gebruikte symbolen in de SchermbeeldCommunicatieDiagrammen (SCD's)



Gebruikers van 'GIRAF' krijgen door middel van **subsysteem** 3 een aantal gereedschappen voor batch-verwerking. Zo bevat dit subsysteem o.a. uitgebreide rapportagefaciliteiten, zoals statistische overzichten omtrent de omvang en samenstelling van de GTD, het afdrukken van een gebruikershandleiding, het afdrukken van een attribuutcodeoverzicht etc. Ook een batch-selectieprogramma, om grote deelbestanden te construeren, behoort tot dit subsysteem.

'GIRAF' wordt aangeboden als **confectiesysteem**. De opzet is echter dusdanig, dat met weinig inspanning wijzigingen kunnen worden aangebracht. Op die manier kan van een **maatconfectiesysteem** worden gesproken. Bepaalde categorieën van gebruikers zullen 'GIRAF' willen (laten) uitbreiden met een 'eigen' on-line systeem (subsysteem 4). De structuur van 'GIRAF' maakt dit mogelijk. In 4.2 wordt hierop nader ingegaan.



Figuur 3b: Voorbeeld van een SchermbeeldCommunicatieDiagram (SCD)

3 ADS/ONLINE

3.1 Inleiding

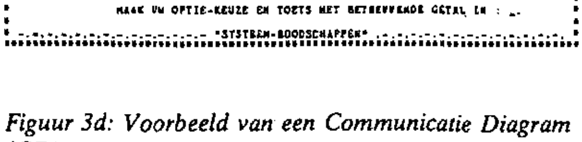
Sinds 1973 is het database-managementsysteem IDMS beschikbaar. IDMS (Integrated Data Management System) is een database-managementsysteem, dat gebaseerd is op het netwerkmodel, zoals dat gedefinieerd is in het 'CODASYL-report' [10].

IDMS wordt zowel in de Verenigde Staten van Amerika als daarbuiten door veel bedrijven en ondernemingen gebruikt als hulpmiddel bij activiteiten op het gebied van de geautomatiseerde gegevensverwerking. Gebruikers die on-line toepassingen construeren gebruiken IDMS in combinatie met een teleprocessing-monitor (kortweg TP-monitor), bijvoorbeeld de TP-monitor CICS/VS van IBM. In 1979 annonceerde de firma Cullinet de TP-monitor IDMS/DC, die geïntegreerd was met IDMS.

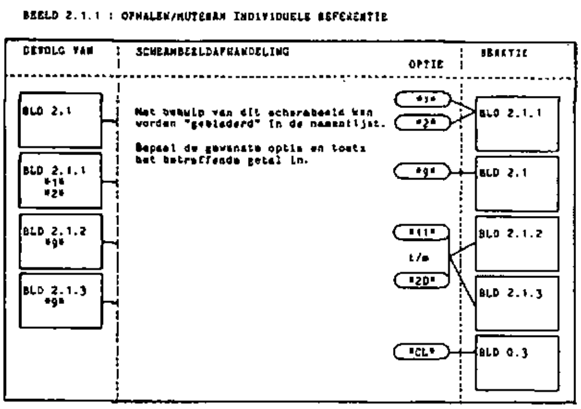
Het ontwikkelen van on-line toepassingen met behulp van een TP-monitor was aanvankelijk een tamelijk moeilijke en moeizame aangelegenheid waarvoor gespecialiseerde programmeurs nodig waren. Aan het eind van de jaren zeventig werd het ontwikkelen van on-line toepassingen wat vereenvoudigd toen de macro-level programmeerstructuren werden vervangen door command-level programmeerstructuren [13].

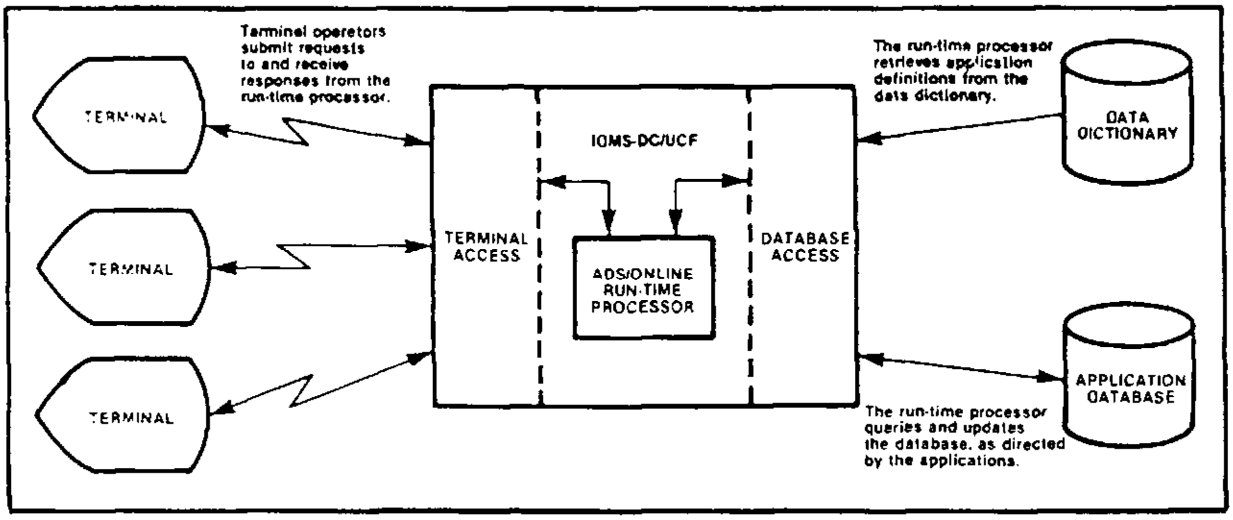
Het aantal on-line toepassingen nam sinds die tijd drastisch toe. Daarom werd gezocht naar hulpmiddelen, die het maken van toepassingsprogramma's verder zouden vereenvoudigen. ADS/ONLINE is zo'n hulpmiddel. Het is als het ware een 'ontwikkeltool'. Met ADS/ONLINE kunnen TP-toepassingen niet alleen sneller worden ontwikkeld, maar ze zijn ook flexibeler m.b.t. het verrichten van aanpassingen. ADS/ONLINE moet gezien worden als een TP-toepassing onder IDMS/DC. Centraal staat het begrip dialoog (zie 3.2). Met behulp van ADS/ONLINE kunnen dialogen worden ontwikkeld. Belangrijk is dat deze dialogen ook op uitvoeringstijd werken onder

Figuur 3c: Voorbeeld van een Schermbeeld Definitie (SD)



Figuur 3d: Voorbeeld van een Communicatie Diagram (CD)





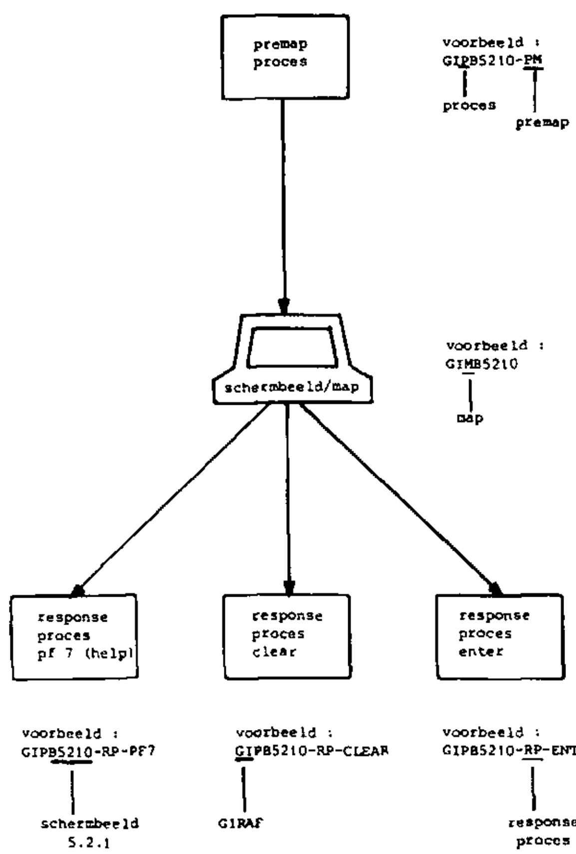
Figuur 4: De omgeving waarbinnen ADS/ONLINE functioneert (Bron: ADS/ONLINE Summary Description)

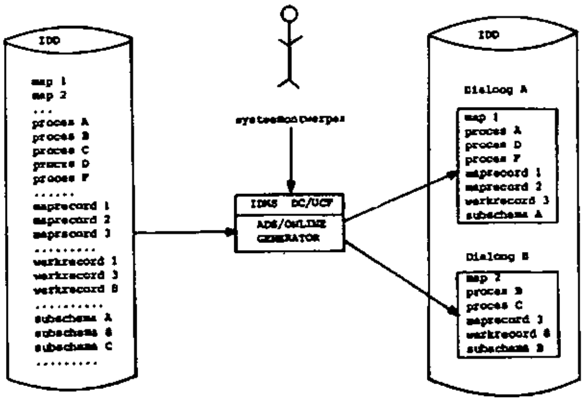
ADS/ONLINE. Figuur 4 laat zien hoe op uitvoeringstijd ADS/ONLINE-dialogen communiceren met terminals en database. In het kader van dit artikel is het niet mogelijk om ADS/ONLINE in extenso te behandelen.
In de volgende paragrafen wordt ingegaan op enkele basisbegrippen van ADS/ONLINE, zoals het begrip dialoog en de samenhang tussen dialogen. Ook wordt aandacht besteed aan de manier, waarop schermbeelden kunnen worden vervaardigd. Het hoofdstuk wordt besloten met een paragraaf waarin een vergelijking wordt gemaakt tussen het ontwikkelen van on-line toepassingen onder ADS/ONLINE, het ontwikkelen van on-line toepassingen m.b.v. conversationele programmering en het ontwikkelen van on-line toepassingen m.b.v. pseudo-conversationele programmering.

3.2 Dialogen

Centraal in ADS/ONLINE staat de **dialoog**. Deze term verdient enige toelichting, omdat de begrippen achter een term nogal eens verschillen [7]. Zo verstaat Craenen [9] onder een dialoog: 'een interactieve vorm van communicatie tussen mens en computer'.
Janssens [20] onderkent drie elementaire dialoogstructuren, namelijk de sequentie, de selectie en de iteratie. Zijn opbouw van dialoogstructuren lijkt voornamelijk geschikt voor conversationele programmering. In een dialoog van ADS/ONLINE staat het **schermbeeld** centraal (zie figuur 5). Een schermbeeld wordt beschreven met behulp van een map. Een **map** is een soort tabel, waarin alle velden van een bepaald schermbeeld met een symbolische naam worden gedefinieerd. De velden bestaan uit tekstvelden en invoer-/uitvoervelden. Deze laatste velden worden gekoppeld aan de velden van één of meer records, de zogenaamde **maprecords**. Deze maprecords zijn gedefinieerd in de Integrated Data Dictionary (IDD) behorend tot IDMS.
Naast schermbeeld (of map) kent een ADS/ONLINE-dialoog een zogenaamd **premap-proces** en **response-processen**. Een premap-proces is een programma, dat nadat de betreffende dialoog geactiveerd is, wordt uitgevoerd voordat de map (het schermbeeld) op het beeldscherm zichtbaar wordt gemaakt. Nadat de gebruiker van de dialoog, die onderdeel is van een of andere toepassing, gereageerd heeft op hetgeen hij op het beeldscherm heeft gezien, wordt een bij de dialoog behorend response-proces gestart. De keuze van het response-proces wordt bepaald aan de hand van de door de gebruiker ondernomen actie. Zo zal het indrukken van de 'enter'-toets een ander response-proces kunnen doen activeren dan het indrukken van de 'clear'-toets. Processen hebben, naast maprecords, vaak werkrecords nodig.
Resumerend kan worden gesteld, dat een ADS/ONLINE-dialoog (doorgaans) bestaat uit: een premap-proces, een map met bijbehorende maprecords, één of meer response-processen en werkrecords. In figuur 5 wordt

Figuur 5: Voorbeeld van een DialoogStructuurDiagram (DSD)





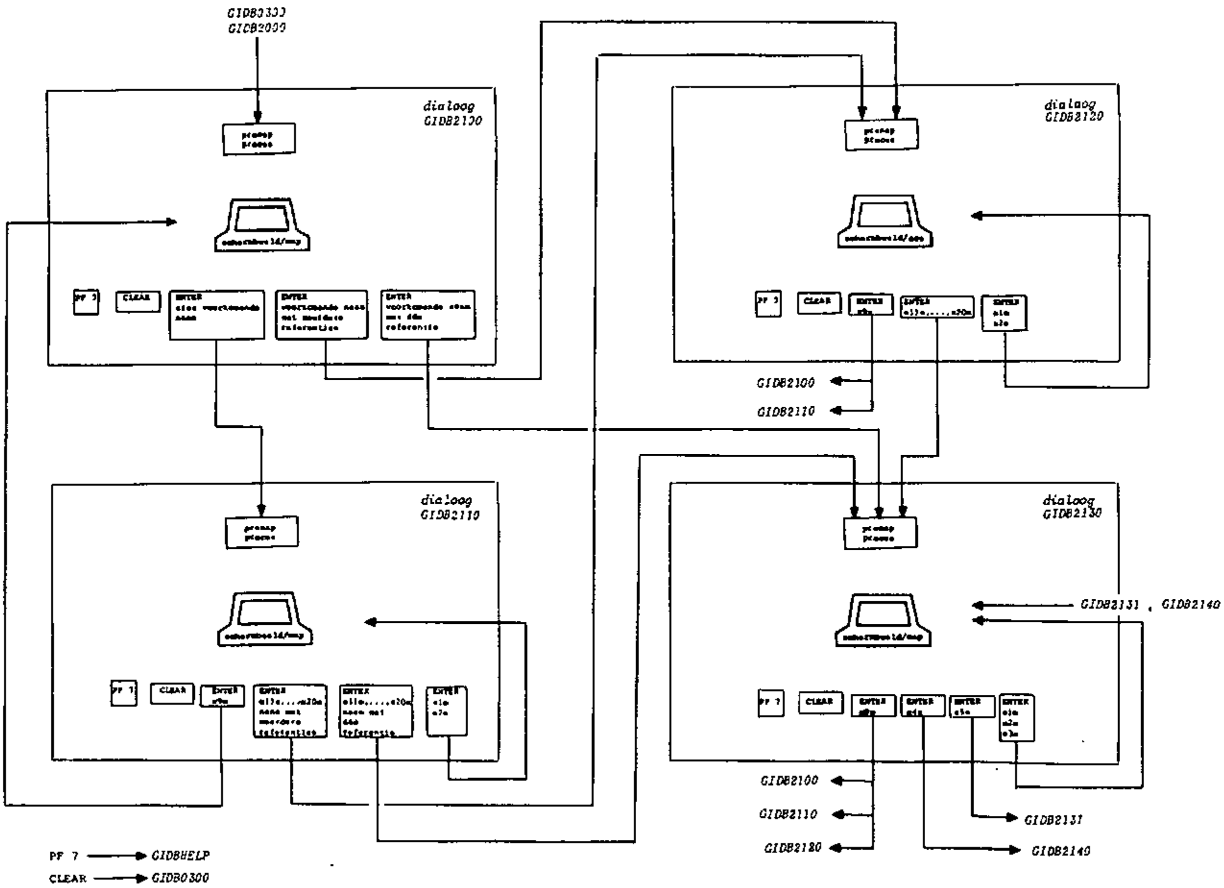
Figuur 6: De ADS/ONLINE-generator (samenknoper)

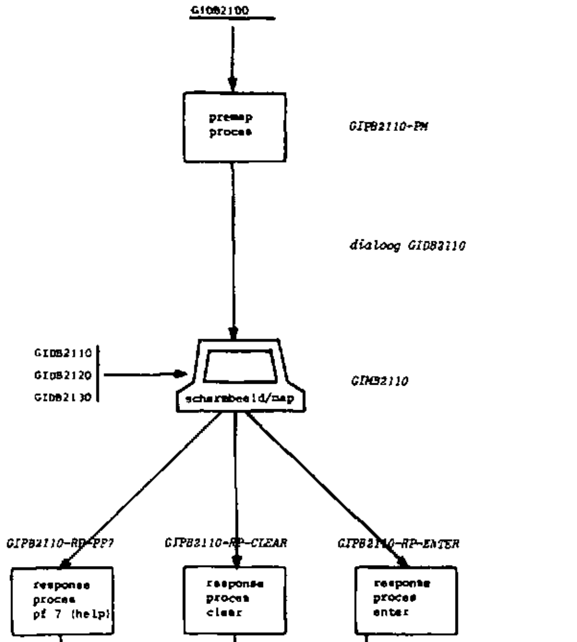
met een DialoogStructuurDiagram (DSD) de samenhang tussen processen en map weergegeven. Maprecords, werkrecords en processen worden in IDD gedefinieerd. De map kan via batch-verwerking worden aangemaakt en in IDD worden geplaatst. Maps kunnen echter ook worden aangemaakt met het Cullinet-produkt On-Line Mapping (OLM). OLM biedt de mogelijkheid om, zoals de naam van het betreffende produkt aangeeft, online maps te definiëren. Op de ervaringen met OLM wordt in 3.4 nader ingegaan. Een dialoog wordt vervaardigd met behulp van de ADS/ONLINE-generator. Deze generator knoopt de juiste ingrediënten samen tot een ADS/ONLINE-dialoog (zie figuur 6). Invoergegevens voor de (on-line) generator zijn map, maprecords, premap-proces, response-processen, werkrecords en subschema.

3.3 Samenhang tussen dialogen

Een dialoog is doorgaans onderdeel van een verzameling dialogen. Deze verzameling dialogen vormt dan het online gedeelte van een toepassing. Hoe kan nu de samenhang tussen de betreffende dialogen worden geregeld? Eén of meer dialogen fungeren als **hoofddialoog/startdialoog**. Nadat een hoofddialoog is geactiveerd, kan een volgende dialoog worden gestart door in een proces een startcommando te geven. Voorbeelden van startcommando's zijn: INVOKE 'GIDB3121', INVOKE GIDB3121-NAAM, LINK 'GIDB3121'. De betreffende dialoognaam kan als literal worden opgegeven (eerste voorbeeld), maar kan ook als een variabele worden opgegeven (tweede voorbeeld). Deze laatste mogelijkheid is voor het systeem 'GIRAF' van zeer groot belang gebleken. In 4.2 zal hierop nader worden ingegaan.
Het is wenselijk om goed inzicht te hebben hoe het verloop tussen de dialogen kan plaatsvinden. Het DialoogCommunicatieDiagram (DCD) (zie figuur 7a) is één van de mogelijkheden om dit te realiseren. Vanuit de response-processen van een dialoog verwijst een pijl naar de dialoog, die door het betreffende response-proces wordt geactiveerd. Met het DCD liggen alle paden, die in een toepassing kunnen worden gevolgd, vast. In de figuren 3a en 3b is aangegeven op welke wijze in het functioneel ontwerp de schermbeelden en het verloop tussen de schermbeelden (m.b.v. SchermbeeldCommunicatie-

Figuur 7a: Voorbeeld van een DialoogCommunicatieDiagram (DCD) (type 1)





Figuur 7b: Voorbeeld van een DialoogCommunicatieDiagram (DCD) (type 2)

Diagrammen (SCD)) wordt vastgelegd. Deze methode met SCD's werd reeds toegepast bij het systeem BIBINFO (1976) en is onafhankelijk van de verdere technische ontwikkeling van het systeem. Opvallend is nu dat de DCD's zeer nauw aansluiten bij de SCD's. Daaruit kan worden geconcludeerd dat met ADS/ONLINE in het technisch ontwerp op een hoger abstractieniveau kan worden gewerkt dan met bijvoorbeeld IDMS/DC of CICS/VS [13,19]. De overgang van functioneel ontwerp naar technisch ontwerp zal soepeler kunnen verlopen. Bovendien kunnen opdrachtgevers, die het functioneel ontwerp kennen, de vorderingen van het technisch ontwerp makkelijker volgen. Het vervaardigen van DCD's, zoals in figuur 7a weergegeven, is bij een on-line programmatuursysteem van enige omvang een nogal tijdrovende bezigheid. Een andere manier om DCD's te maken is het toevoegen van de communicatiegegevens aan het DialoogStructuurDiagram (DSD). In figuur 7b is een en ander weergegeven.

3.4 OnLine Mapping (OLM)

OnLine Mapping (OLM) is geen direct onderdeel van ADS/ONLINE, maar is een hulpmiddel van IDMS/DC. Met OLM is het mogelijk om on-line maps te vervaardigen. De ontwerper krijgt een 'blanco schermbeeld' te zien. Hij kan onmiddellijk het schermbeeld gaan ontwerpen door tekstvelden en invoer-/uitvoervelden te definiëren. OLM heeft een aantal conventies, waarmee een en ander bewerkstelligd kan worden. Een voordeel is dat de lay-out van het schermbeeld direct kan worden bekeken en desgewenst kan worden aangepast. Dit is vooral interessant bij de bouw van prototype (T) in de definitiestudie. In 4.4 wordt hierop nader ingegaan. Nadat tekstvelden en invoer-/uitvoervelden op het schermbeeld zijn gedefinieerd krijgt de ontwerper de mogelijkheid om aan alle velden, zowel de tekstvelden als invoer-/uitvoervelden, attribuutgegevens toe te kennen zoals protected/unprotected, lichtintensiteit normaal/helder/donker etc. De invoer-/uitvoervelden moeten worden gekoppeld aan een veld van een maprecord. Nadat alle velden voorzien zijn van attribuutgegevens kan de map worden gegenereerd en in IDD worden opgenomen. Daarmee is de map beschikbaar voor het maken van een ADS/ONLINE-dialoog. OLM is op die manier een prettig hulpmiddel om maps te vervaardigen voor ADS/ONLINE-dialogen.

De opgedane ervaringen met OLM geven aanleiding tot het maken van enkele opmerkingen:

a. In diverse TP-monitoren is het niet mogelijk om bij het koppelen van invoer-/uitvoervelden, die tot de map behoren, aan velden van werkrecords (de maprecords) gebruik te maken van arrays. OLM biedt de mogelijkheid tot het werken met een-dimensionale arrays. Dit maakt de programmering eenvoudiger. Bij 'GIRAF' bestond bij diverse maps de behoefte om te kunnen werken met twee-dimensionale arrays. Helaas is het, werkend met OLM, niet mogelijk om met twee-dimensionale arrays te werken.
b. Binnen 'GIRAF' wordt op diverse plaatsen gewerkt met maps, die 'dynamisch samengesteld' worden. Op één punt was het gewenst de lengte van een veld in bepaalde situaties te verkleinen. Hoewel attribuutgegevens van een veld dynamisch kunnen worden gewijzigd, is het helaas niet mogelijk een veld (tijdelijk) korter te maken.
c. Nadat met OLM een map vervaardigd is, ligt de koppeling tussen één of meer maprecords en de map vast. Een probleem is nu dat het loskoppelen van een maprecord van de map niet mogelijk is zonder eerst de map te elimineren. Aangezien dit doorgaans nogal wat werk met zich meebrengt blijft het niet meer gebruikte maprecord vaak gekoppeld aan de map. Indien er veel wijzigingen optreden komt een en ander de overzichtelijkheid niet ten goede.

Geconcludeerd kan worden dat OLM een goed produkt is om maps on-line te vervaardigen. Daarom verdient dit produkt het dat aan bovenstaande opmerkingen de nodige aandacht wordt geschonken.

3.5 ADS/ONLINE versus (pseudo-)conversationele programmering

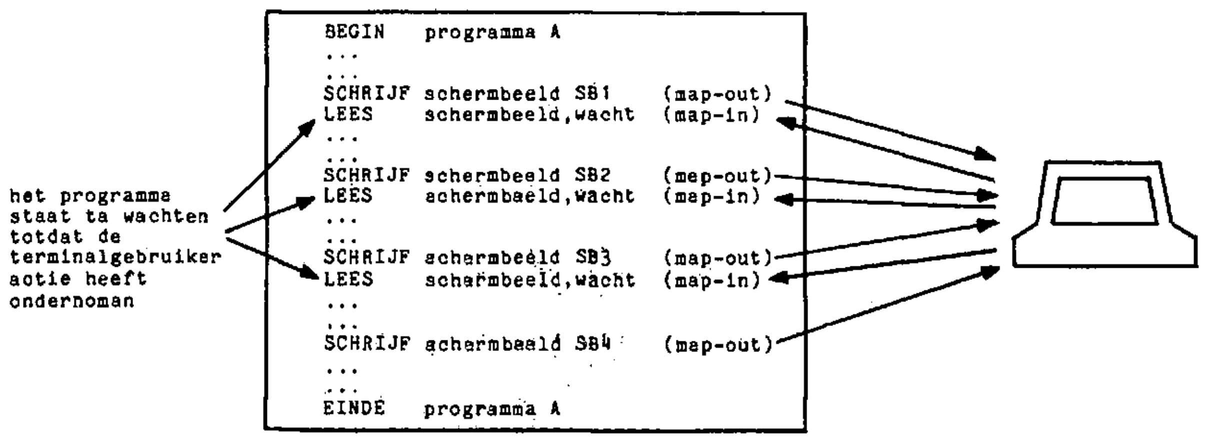
Bij on-line informatiesystemen/programmatuursystemen [14] is sprake van onvertraagde (real-time) verwerking van logische eenheden te verwerken gegevens. Deze vorm van gegevensverwerking staat bekend als postgewijze/transactiegewijze gegevensverwerking (transaction processing). De postgewijze gegevensverwerking vindt plaats met behulp van conversaties. Onder een conversatie kan worden verstaan: een vraag- en antwoordspel tussen een gebruiker en een computer met behulp van een beeldscherm. Voor het laten functioneren van on-line postgewijze gegevensverwerkende systemen zijn twee ontwerpstrategieën beschikbaar: conversationele programmering en pseudo-conversationele programmering. Ook een combinatie van beide ontwerpstrategieën is mogelijk. Allereerst wordt nu een korte beschrijving van conversationele programmering en pseudo-conversationele programmering gegeven. Daarna wordt de relatie met ADS/ONLINE aangegeven.

3.5.1 *Conversationele programmering*
Bij conversationele programmering worden meerdere conversaties tussen computer en gebruiker in één programma verwerkt (zie figuur 8a). Dit betekent dat het programma de invoer van de gebruiker inleest, deze invoer verwerkt en het resultaat op het beeldscherm plaatst. Daarmee is (een deel van) de post of de transactie met behulp van een conversatie verwerkt. Vervolgens is de gebruiker aan de beurt om gegevens in te toetsen. Dit kan echter enige tijd vragen, soms enkele seconden maar ook weleens enkele minuten. Gedurende deze reactietijd van de gebruiker staat het programma te wachten. Tijdens dat wachten is het programma als het ware 'slapend' maar vraagt wel aandacht van de TP-monitor. Nadat de gebruiker de 'enter'-toets, de 'clear'-toets of een functietoets heeft ingedrukt kan het programma weer in actie komen. Deze procedure herhaalt zich totdat de gebruiker er voor zorgt dat het programma kan worden beëindigd door via de invoergegevens te laten weten dat hij zijn sessie wil beëindigen. Aan veel programmatuursytemen wordt de eis gesteld dat de responsietijden < 4 seconden moeten zijn. Dit betekent dat bij conversationele programmering het programma slechts een zeer klein deel van de verblijftijd actief is.
Voor de komst van TP-monitoren zoals CICS/VS en IDMS/DC bestond een on-line programmatuursysteem vaak uit slechts één conversationeel programma. Een voorbeeld daarvan is het systeem 'BIBINFO' [12]. Dit on-line information retrieval systeem is in 1976 gemaakt. Het on-line gedeelte van dit systeem bestaat uit één groot COBOL-programma dat enkele duizenden instructies omvat. Met behulp van een tiental subroutines, zoals READSCR, WRTSCR etc., wordt de communicatie met de TP-monitor TSO (Time Sharing Option van IBM) geregeld om schermbeelden te maken, te wijzigen, te versturen en in te lezen. Dergelijke conversationele programma's lijken veel op grote batch-programma's. De ervaring heeft geleerd dat geroutineerde batch-programmeurs veel makkelijker kunnen overschakelen naar on-line programmering met behulp van de conversationele programmeringsstrategie dan met pseudo-conversationele programmering.

3.5.2 *Pseudo-conversationele programmering*
Conversationele programma's zijn slechts een zeer klein deel van de verblijftijd actief. Het doel van pseudo-conversationele programmering is nu om dergelijke programma's als het ware in stukjes te knippen. Ieder stukje handelt een (deel van een) transactie af. Het betreffende programma(deel) wordt pas 'opgeroepen' als een gebruiker op de 'enter'-toets, de 'clear'-toets of een functietoets heeft gedrukt (zie figuur 8b). Nadat de invoergegevens zijn verwerkt wordt het resultaat naar het beeldscherm verstuurd. Direct daarna verdwijnt het programma(deel) van het toneel. Het voordeel van deze werkwijze is dat er geen 'slapende' programma's in het geheugen aanwezig zijn. In plaats van één groot programma bestaat bij pseudo-conversationele programmering de applicatie uit een groot aantal kleine (en doorgaans eenvoudige) programma's. Deze programma's zijn als het ware subprogramma's die aan de TP-monitor worden toegevoegd. De TP-monitor verzorgt o.a. alle invoer- en uitvoeropdrachten [13, 19] en kan worden beschouwd als een soort besturingssysteem voor deze programma's. Voor het operating systeem is de TP-monitor echter een gewoon applicatieprogramma. Het vervaardigen van een programmatuursysteem met behulp van pseudo-conversationele programmering vergt een wat andere denktrant dan het maken van grote batch-programma's. Ervaren batch-programmeurs die moeten overschakelen naar pseudo-conversationele programmering hebben het gevoel dat ze niet zelf meer alles in de hand hebben. Het parallel kunnen werken van diverse programma's en de communicatie tussen alle kleine programma's zorgen in het begin voor een onzeker gevoel. Nadat deze programmeurs enkele programma's hebben gemaakt verdwijnt dat gevoel langzaam maar zeker.

3.5.3 *ADS/ONLINE pseudo-conversationeel*
ADS/ONLINE werkt, zoals aangegeven in 3.2, met dialogen. In een dialoog staat het schermbeeld (map) centraal. Naast het schermbeeld kent een dialoog een premap-proces en response-processen. Bij pseudo-conversationele programmering staat het proces, dat de transactie moet verwerken, centraal. Het programma start met het lezen van het schermbeeld (map-in), verwerkt de gegevens en verstuurt een schermbeeld naar het beeldscherm (map-out). ADS/ONLINE heeft dus een duidelijk afwijkende werkwijze voor de systeemontwerper/programmeur dan de pseudo-conversationele werkwij-

Figuur 8a: Conversationele programmering



ze. Toch werkt ADS/ONLINE intern pseudo-conversationeel. In dit artikel wordt niet verder ingegaan op de technische details van een en ander. Vastgesteld kan worden dat met ADS/ONLINE de systeemontwerper/programmeur op een hoger abstractieniveau kan werken dan met een pseudo-conversationele werkwijze (zie 3.3). Desondanks kan er worden geprofiteerd van de voordelen van de pseudo-conversationele werkwijze.

4 ADS/ONLINE IN RELATIE TOT DE TOEPASSING

4.1 Inleiding

In dit hoofdstuk wordt ingegaan op de ervaringen die zijn opgedaan met ADS/ONLINE. Er wordt verslag gedaan van een ontwikkeld cassettesysteem. Met dit cassettesysteem kunnen de flexibiliteitsaspecten open-ended en adaptability worden gerealiseerd. Verder wordt er ingegaan op de ervaringen die zijn opgedaan met multiple databases en prototyping.

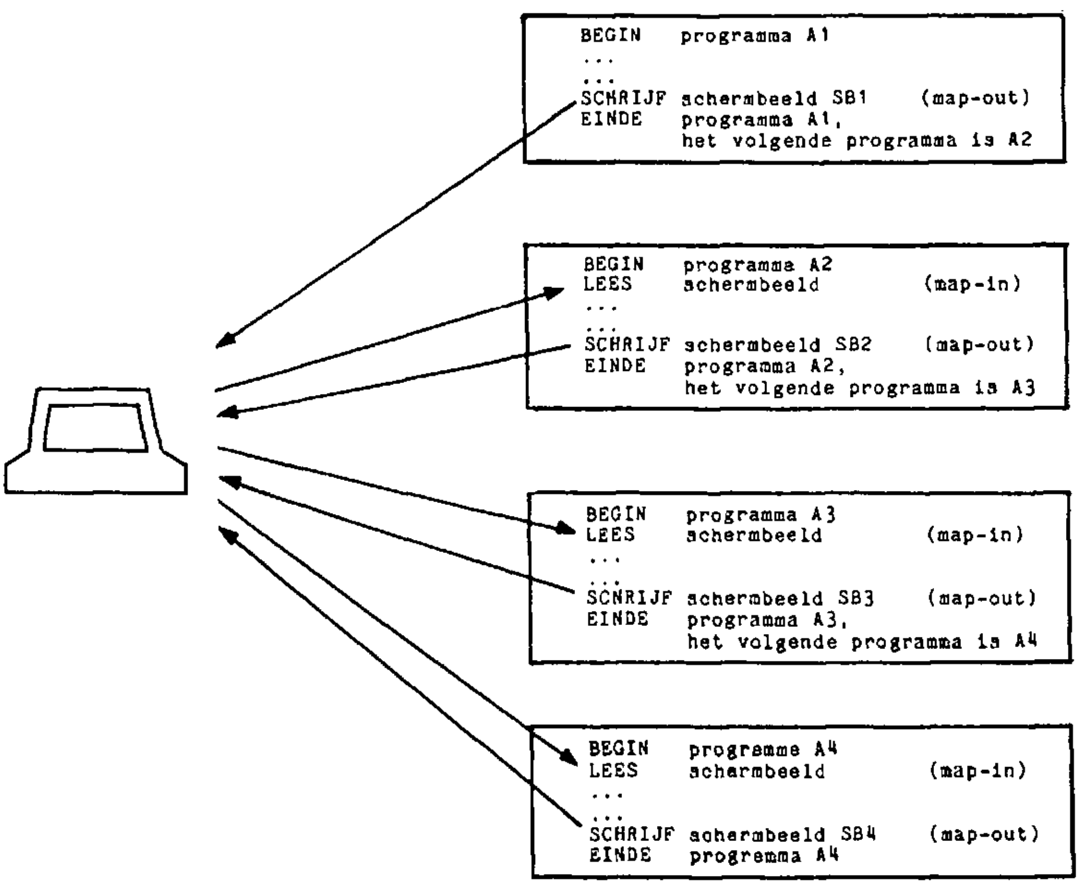
4.2 Communicatie tussen dialogen

Zoals in 3.3 beschreven is vindt de communicatie tussen dialogen plaats door middel van startcommando's in de response-processen van de dialogen. Als voorbeelden van zo'n startcommando zijn o.a. gegeven: INVOKE 'GIDB3121' en INVOKE GIDB3121-NAAM. In het laatste voorbeeld staat de naam van de dialoog die wordt gestart in de variabele GIDB3121-NAAM. Deze mogelijkheid is voor het programmatuursysteem 'GIRAF' van groot nut gebleken. Binnen 'GIRAF' gebruiken alle response-processen startcommando's met een variabele als operand. De naam van de dialoognaamvariabele is gelijk aan de naam van de dialoog die volgens het DialoogCommunicatieDiagram (DCD) (zie figuren 7a,b) in de betreffende situatie moet worden gestart. De naam van de dialoognaam-variabele wordt gevolgd door de suffix '-NAAM'. Waarom is een en ander nu zo belangrijk? Binnen het programmatuursysteem 'GIRAF' is een cassettesysteem ontwikkeld. Dit houdt in dat, nadat de gebruiker van een GIRAF-Toepassings-Database (GTD) zich heeft gemeld, alle dialoognaam-variabelen door het systeem voor die gebruiker worden gevuld. Allereerst wordt nu ingegaan op de vraag waarom? Daarna zal aan de orde komen hoe een en ander geregeld is.

4.2.1 *De motivatie van het cassettesysteem*

De volgende motieven hebben een rol gespeeld bij de ontwikkeling van het cassettesysteem:

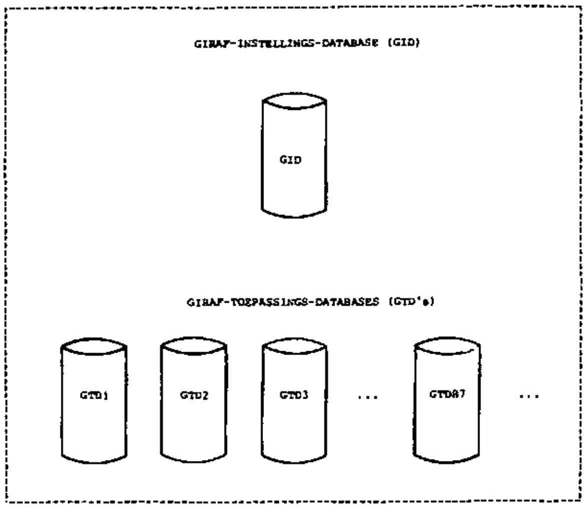
a. 'GIRAF' is een algemeen toepasbaar programmatuursysteem en wordt gebruikt voor veel toepassingen. 'GIRAF' wordt aangeboden als confectiesysteem. Eén van de aan 'GIRAF' gestelde eisen was dat het met weinig inspanning moet kunnen worden aangepast. Op die manier kan 'GIRAF' voor sommige gebruikers fungeren als maatconfectiesysteem. Het op maat maken van 'GIRAF' voor een specifieke gebruiker kan nu worden gerealiseerd door de betreffende dialogen te kopiëren, van een andere naam te voorzien en aan te passen.



Figuur 8b: Pseudo-conversationele programmering

Nadat de betreffende gebruiker zich heeft aangemeld bij 'GIRAF', wordt er voor gezorgd dat in de betreffende dialoognaam-variabelen de afwijkende dialoognaam komt te staan. De oproepende dialoog gebruikt in het startcommando de dialoognaam-variabele. De inhoud van deze variabele bepaalt welke dialoog wordt gestart.

b. Bij de TH-Delft werken en studeren ook buitenlanders. 'GIRAF' moet daarom kunnen werken met andere talen. De gebruiker geeft bij het aanmelden met behulp van een taalcode aan in welke taal hij wil werken. Het cassettesysteem zorgt er niet alleen voor dat de namen van de (gebruikersafhankelijke) dialogen worden geladen, maar zorgt er tevens voor dat, afhankelijk van de opgegeven taalcode, de juiste (taalafhankelijke) dialoognamen worden geladen. De gebruiker kan daarna werken met 'zijn' maatconfectiesysteem in de door hem gewenste taal.



Figuur 9a: Structuur van de GIRAF-DATABASE

4.2.2 *De werking van het cassettesysteem*

Een GIRAF-database bestaat (zie figuur 9a) uit een GIRAF-INSTELLINGS-DATABASE (GID) en voor iedere toepassing een GIRAF-TOEPASSINGS-DATABASE (GTD). De GID bevat een recordtype GI-DIANMALG-REC (zie figuur 9b). Een record-occurrence van dit recordtype bevat een dialoognaam als entrynaam en vervolgens tien dialoognaam-substituten. Afhankelijk van de taalcode wordt één van deze dialoognaamsubstituten gesubstitueerd voor de betreffende dialoognaam.

Voorbeeld:

GIDB3121	(entrynaam)	
GIDB3121	(Ned. dialoog met hoofdletters)	(subst. 1)
GIDN3121	(Ned. dialoog met kleine letters)	(subst. 2)
GIDE3121	(Engelstalige dialoog)	(subst. 3)
GIDF3121	(Franstalige dialoog)	(subst. 4)
. . . .		
GIDH3121	(Hongaarstalige dialoog)	(subst. 10)

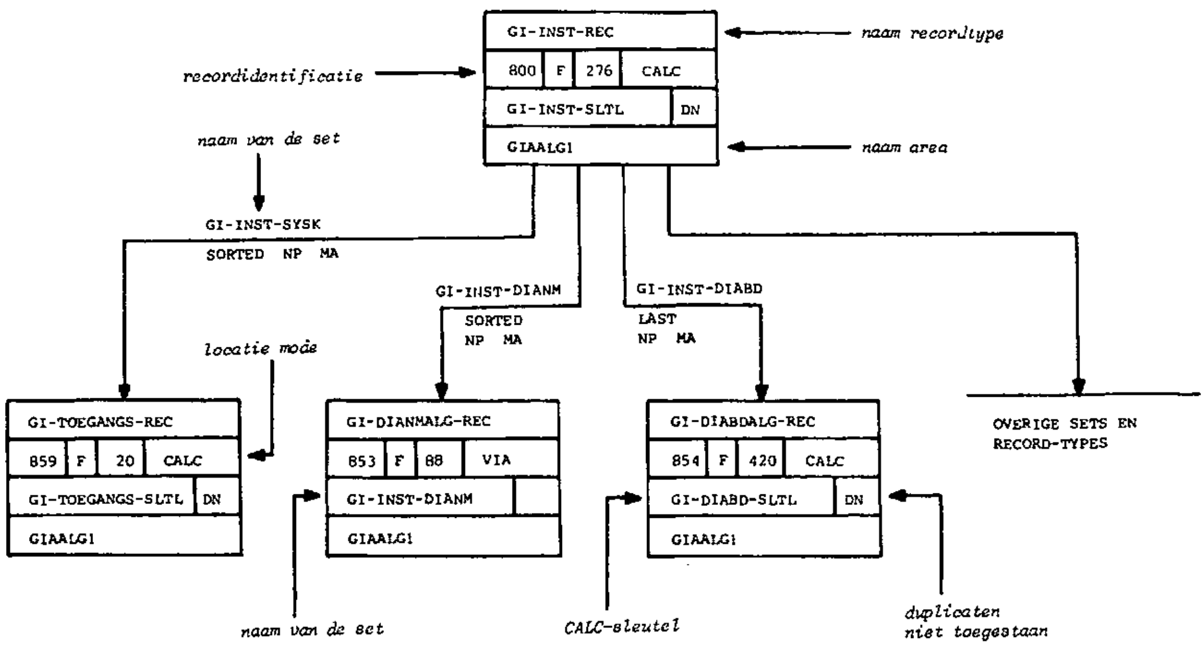
Ten behoeve van een GTD kunnen afwijkende dialogen zijn ontwikkeld. De GTD bevat een recordtype GI-DIANMAFW-REC (zie figuur 9c). De structuur van een record-occurrence is hetzelfde als de structuur van een record-occurrence van recordtype GI-DIANMALG-REC.

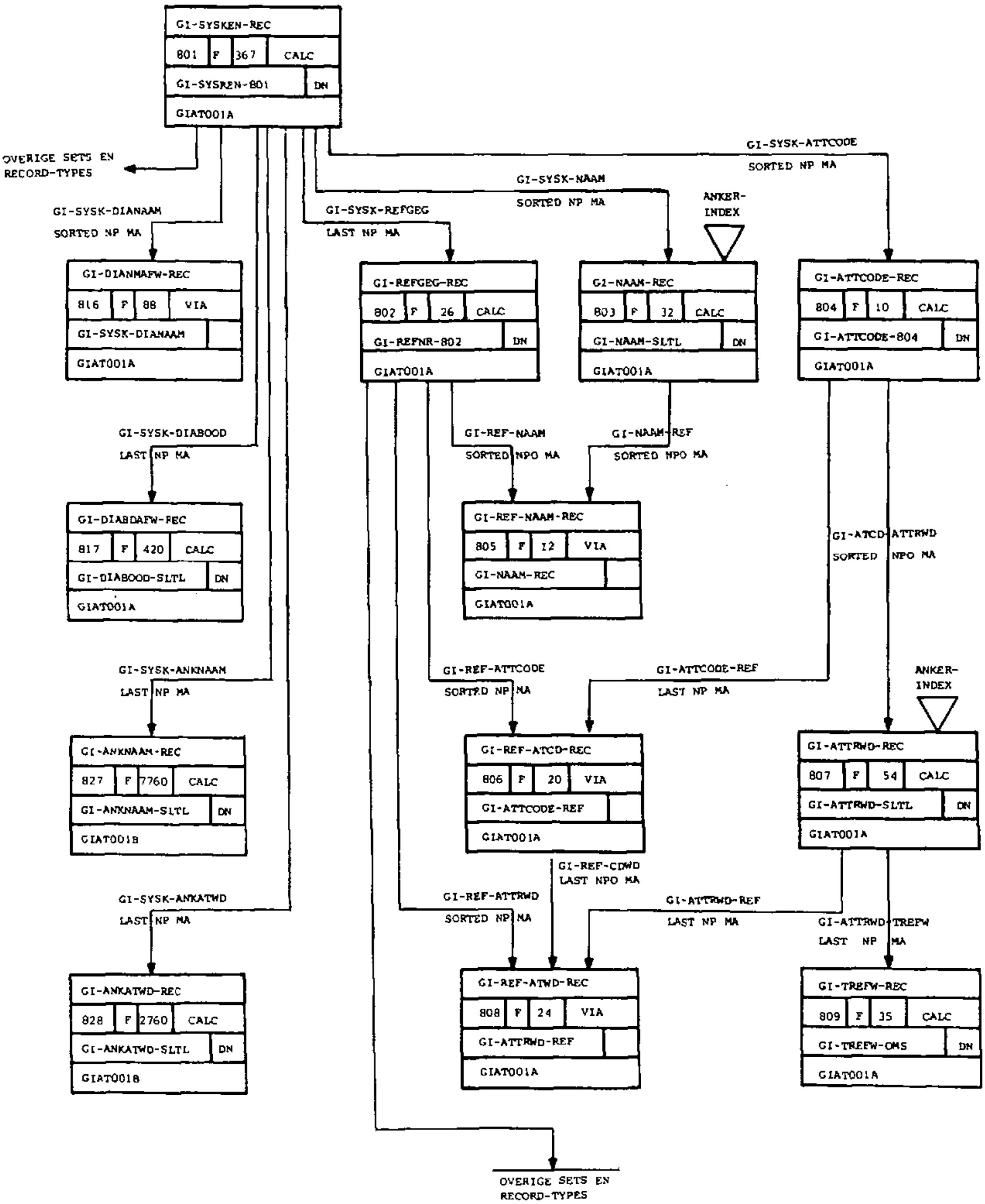
Voorbeeld:

GIDB3121	(entrynaam)
MADB3121	(afw. Ned. dialoog met hoofdletters)
MADN3121	(afw. Ned. dialoog met kleine letters)
MADE3121	(afwijkende Engelstalige dialoog)
MADF3121	(afwijkende Franstalige dialoog)
. . . . .	
MADH3121	(afwijkende Hongaarstalige dialoog)

Vaak zullen afwijkende dialogen maar in één taal beschikbaar zijn. In dat geval kan de inhoud van de betreffende record-occurrence bijvoorbeeld zijn:

Figuur 9b: Structuur van de GIRAF-Instellings-Database (GID)

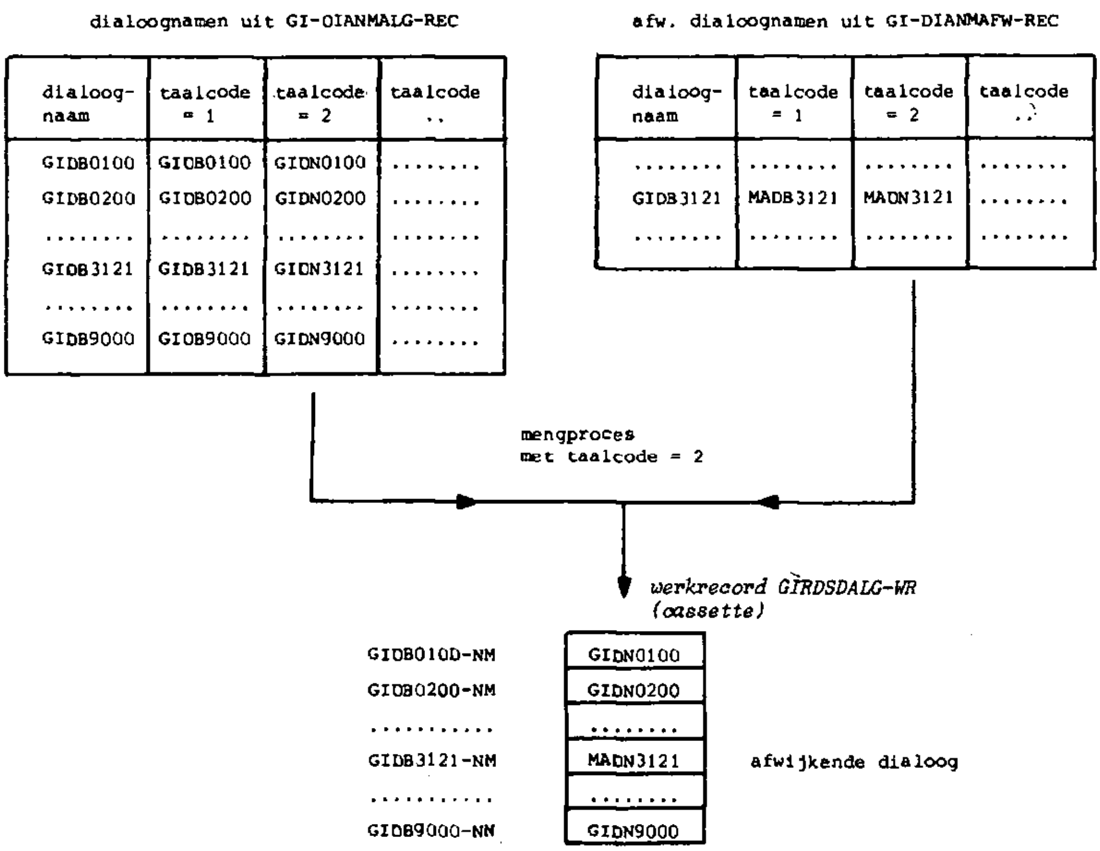




Figuur 9c: Structuur van de GIRAF-Toepassings-Database (GTD)

GIDB3121 (entrynaam)
MADN3121 (afw. Ned. dialoog met kleine letters)
MADN3121 (afw. Ned. dialoog met kleine letters)
MADN3121 (afw. Ned. dialoog met kleine letters)
MADN3121 (afw. Ned. dialoog met kleine letters)
. . . . .
MADN3121 (afw. Ned. dialoog met kleine letters)

Het cassettesysteem bestaat nu uit het mengen van de record-occurrences van recordtype GI-DIANMALG-REC met de record-occurrences van recordtype GI-DIANMAFW-REC (zie figuur 10). Het mengproces houdt in dat afhankelijk van de taalcode een dialoognaam-substituut uit een record-occurrence van GI-DIANMALG-REC in het werkrecord GIRDSDALG-WR wordt geplaatst, tenzij er voor de betreffende dialoognaam een record-occurrence van recordtype GI-DIANMAFW-REC bestaat. In dat geval wordt afhanke-



Figuur 10: De werking van het cassettesysteem

lijk van de taalcode een substituut van de afwijkende dialoognaam in het werkrecord (de cassette) geplaatst. Na afloop van het mengproces bevat het werkrecord GIRDSDALG-WR alle dialoognamen die voor de betreffende gebruiker op dat moment relevant zijn. Zowel met afwijkende dialogen als met de gewenste taalcode is rekening gehouden. Het werkrecord GIRDSDALG-WR wordt aan alle dialogen gekoppeld zodat alle dialogen over de voor de gebruiker relevante dialoognamen kunnen beschikken.
Iedere dialoog bevat processen. Deze processen kunnen mededelingen naar het beeldscherm zenden. Met name het zogenaamde $-MESSAGE-veld speelt hierbij een belangrijke rol. Afwijkende dialogen produceren vaak ook afwijkende mededelingen. Bovendien moeten de processen onafhankelijk zijn van de door de gebruiker opgegeven taalcode. Om die redenen zijn alle mededelingen buiten de processen gehouden en op soortgelijke wijze als de dialoognamen in de GID en GTD opgeslagen. Ook met betrekking tot de mededelingen is een cassettesysteem ontwikkeld. Dit cassettesysteem vult de cassette (het werkrecord) niet direct nadat de gebruiker zich heeft gemeld, maar werkt op procesniveau. Ieder proces van een dialoog dat een mededeling wil versturen bepaalt eerst via het cassettesysteem wat de op dat moment relevante mededeling is.

4.3 Multiple databases

Verwacht wordt dat 'GIRAF' binnen de TH-Delft ten behoeve van tientallen informatiesystemen zal worden gebruikt. De toepassingen kunnen sterk variëren. Zo zijn er enerzijds individuele wetenschappers die hun literatuurreferenties in een GIRAF-TOEPASSINGS-DATABASE (GTD) opslaan en anderzijds toepassingen die werken met tienduizenden referenties. Gezien het grote aantal toepassingen en het uiteenlopende karakter van de toepassingen is gezocht naar een mogelijkheid om alle GTD's fysiek te scheiden. Het vervaardigen van één grote database werd niet zinvol geacht. In het laatste geval zouden de gebruikers met kleine GTD's zeer vermoedelijk te veel worden gehinderd door gebruikers die 'GIRAF' gebruiken in een informatiesysteem dat dagelijks gedurende lange tijd met (zeer) veel referenties beschikbaar moet zijn. Door de GTD's fysiek te scheiden kan de verdeling van de GTD's over de verza-

Figuur 11: Voorbeeld van een Database Name Table (DNT)

DBNAME	formeel subschema		actueel subschema
GIGTD001	GIVT001A	maps to	GIVT001A
GIGTD002	GIVT001A	maps to	GIVT002A
GIGTD003	GIVT001A	maps to	GIVT003A
....	....	....	....
GIGTD087	GIVT001A	maps to	GIVT087A
....	....	....	....

meling schijven die het Rekencentrum van de TH-Delft bezit, beter worden geregeld. Een tweede belangrijk argument is de beheersbaarheid van de GTD's. Door een fysieke scheiding van de GTD's zal het eenvoudiger zijn om één of meer GTD's te reorganiseren. Van sommige GTD's zal vaker een back-up moeten worden gemaakt dan van andere GTD's. Ook dat is simpeler te realiseren als de GTD's gescheiden zijn.

Om een en ander te realiseren zou in het schema dat de database beschrijft een recordtype opgenomen kunnen worden waarvan een record-occurrence de 'owner' is van alle referenties van een toepassing. Het 'CODASYL-report' [10] staat toe dat record-occurrences van één recordtype in verschillende areas worden geplaatst. Door nu ieder record-occurrence in een aparte area te plaatsen en vervolgens aan iedere area een aparte file te koppelen zou de wens van scheiding van alle toepassingen gerealiseerd kunnen worden. IDMS staat echter niet toe dat record-occurrences van één recordtype in verschillende areas worden geplaatst zodat deze oplossing niet gerealiseerd kon worden. Gelukkig biedt IDMS een andere oplossing, die recentelijk ook binnen ADS/ONLINE kan worden gebruikt, namelijk de zogenaamde DBNAME-opdracht. Met behulp van de DBNAME-opdracht kan een programma dat met subschema A is vertaald op uitvoeringstijd worden gekoppeld aan subschema B. Om een en ander mogelijk te maken moet binnen IDMS een zogenaamde Database Name Table (DNT) zijn gedefinieerd (zie figuur 11). De DNT bevat voor 'GIRAF' een aantal ingangen (entries). Per ingang zijn gedefinieerd: een DBNAME, een formeel subschema en een actueel subschema. Binnen een proces van ADS/ONLINE kan met behulp van de DBNAME-opdracht het formele subschema worden 'vervangen' door het betreffende actuele subschema. De opdracht MOVE 'GIGTD087' TO DBNAME zorgt er voor dat via ingang GIGTD087 de DNT wordt benaderd. In de DNT staat aangegeven dat het formele subschema GIVT001A moet worden vervangen door het actuele subschema GIVT087A. Indien er geen DBNAME-opdracht wordt gegeven, dan is het formele subschema bij verstek het actuele subschema. Opgemerkt moet worden dat de formele en actuele subschema's alleen mogen verschillen v.w.b. de page ranges en files. Recordtype- en setbeschrijvingen moeten hetzelfde zijn.

Binnen 'GIRAF' is het zo geregeld dat de DBNAME van een toepassing is opgeslagen in het betreffende toegangsrecord GI-TOEGANGS-REC van de GID (zie figuur 9b). Nadat de gebruiker zich heeft gemeld wordt er met behulp van de DBNAME uit het betreffende GI-TOEGANGS-REC voor gezorgd dat het juiste actuele subschema (en dus ook de juiste database van de multiple databases) 'in stelling' wordt gebracht.

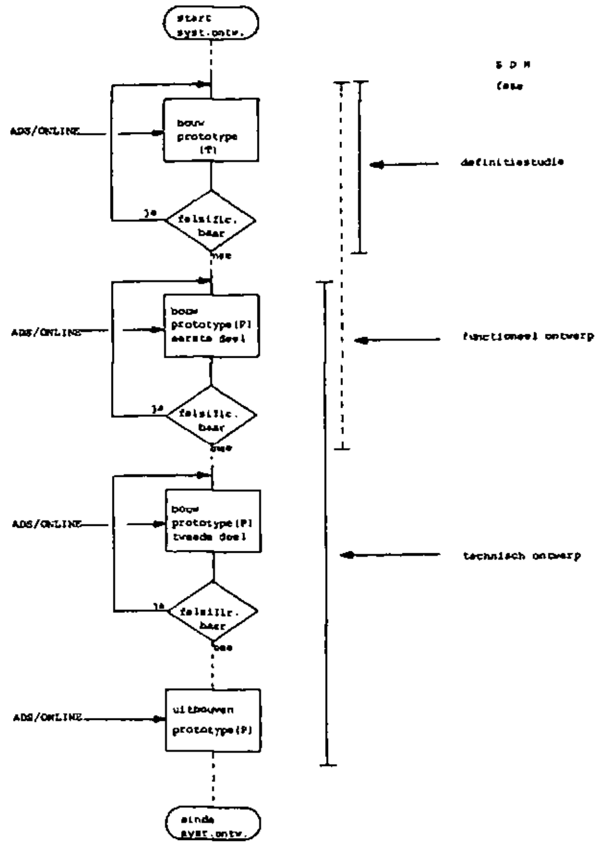
Het werken met bovengenoemde DNT levert een ongewenst neveneffect op. In IDMS is het namelijk mogelijk een relatie tussen een programma en subschema vast te leggen. De Data Base Administrator (DBA) legt die relatie vast om er voor te zorgen dat het betreffende programma uitsluitend met dat subschema kan werken. Het vertalen van het programma met een ander subschema is dan niet mogelijk. Deze koppeling is dus uniek. Het is niet mogelijk aan een programma meerdere subschema's te koppelen. Via het DBNAME-mechanisme kunnen, weliswaar via een omweg, wel meerdere subschema's aan een programma worden gerelateerd. Het is echter gebleken dat gebruik van het DBNAME-mechanisme alleen mogelijk is als de DBA geen vast subschema aan het programma koppelt. Dit betekent dat het betreffende programma eventueel door een programmeur met een ander subschema kan worden vertaald. Gebruik van het DBNAME-mechanisme gaat dus ten koste van een stukje beveiliging. Het is wenselijk om ook bij gebruik van het DBNAME-mechanisme een koppeling tussen een programma en een vast (formeel) subschema mogelijk te maken.

4.4 **Prototyping**

Bij de beschrijving van ADS/ONLINE [1] wordt aangegeven dat dit ontwikkeltool een goed hulpmiddel is om prototyping-activiteiten te ondersteunen. In deze paragraaf worden de ervaringen die zijn opgedaan met prototyping-activiteiten samengevat. Bij de beschouwingen zullen de artikelen van Sol [23] en Van Beek [5] worden betrokken. Van Beek stelt dat een werkwijze als SDM niet bruikbaar is voor een op prototyping gefundeerde systeemontwikkeling. Hij is van mening dat prototyping vooral geschikt is voor systeemontwikkelingsmethoden die zijn gebaseerd op de 'data analyse'-methode. In zijn artikel beschrijft Van Beek de Prototyping Development Methodology (PDM). Deze methode is vooral geschikt voor projecten waar de informatiebehoefte nogal onduidelijk is, zoals MIS en DSS. Van Beek concludeert dat bij administratieve automatisering de informatiebehoefte redelijk goed bekend is en daarom de baten van prototyping minder groot zijn. Bij de ontwikkeling van de systemen 'GIRAF' en 'GEMS' is met behulp van ADS/ONLINE op bescheiden schaal gebruik gemaakt van prototyping. Daarbij kunnen twee niveaus worden onderscheiden (zie figuur 12):

4.4.1 *Prototype (T)*

In de definitiestudie is met ADS/ONLINE een aantal dialogen alsmede de communicatie tussen deze dialogen geconstrueerd. De gegevens die een rol speelden in dit prototype waren gefingeerd. Het prototype bevatte geen database-activiteiten. Met het prototype werd beoogd de eindgebruiker/opdrachtgever op zeer korte termijn, enkele dagen, een aanschouwelijke indruk te geven over de wijze waarop zijn informatievoorziening gerealiseerd zou kunnen worden. Dit prototype had een tijdelijk karakter en wordt kortweg prototype (T) genoemd. Het heeft ook tijdens het functioneel ontwerp een rol gespeeld. Bij de systemen 'BIBINFO' (1976) [12], 'Aanvraagsysteem' (1980) [13], 'GIRAF' (1984) [14] en 'GEMS' (1984/1985) [16] heeft namelijk het ontwerpen van de schermbeelden en de communicatie tussen de schermbeelden plaatsgevonden tijdens de fase functioneel ontwerp. Dit in tegenstelling tot SDM, die aangeeft dat het ontwerpen van schermbeelden in de fase technisch ontwerp moet plaatsvinden. Persoonlijke opvattingen van de ontwerper, die aansluiten bij het motto (Van Rees [22]): 'De ontwerper ontwerpt en niet de methode', speelden hierbij de belangrijkste rol. Het beoordelen van op papier vastgelegde schermbeeld(dialog)en door eindgebruikers is een moeilijke zaak. Gebleken is dat het prototype (T) in deze fase een rol kan spelen. Door demonstraties met het prototype (T) is de eindgebruiker beter in staat om de op papier vastgelegde schermbeeld(dialog)en te kunnen beoordelen en eventueel te falsificeren.



Figuur 12: ADS/ONLINE in relatie tot prototyping en SDM

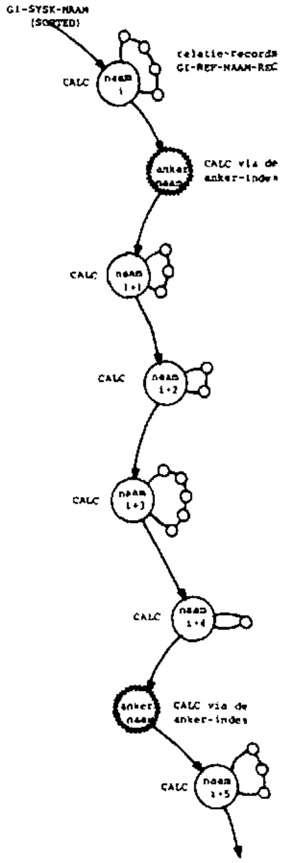
4.4.2 *Prototype (P)*
Een van de eerste activiteiten die in de fase technisch ontwerp is verricht was het bouwen van het prototype (P). Alle schermbeelden en schermbeelddialogen werden met behulp van ADS/ONLINE vervaardigd. De letter P geeft aan dat het een 'permanent prototype' betreft. Dit permanente karakter betekent dat het prototype in een latere fase wordt uitgebouwd tot een onderdeel van het gewenste informatiesysteem. De bouw van prototype (P) is uiteraard veel meer werk dan de bouw van een prototype (T). Opgemerkt kan nog worden dat, indien er voldoende menskracht beschikbaar is, reeds tijdens de fase functioneel ontwerp een gedeelte van prototype (P) kan worden gebouwd. Dit deel van prototype (P) kan dan de rol van het prototype (T) in het functioneel ontwerp overnemen. Ook bij het prototype (P) is gewerkt met gefingeerde gegevens zonder database-activiteiten. Na het realiseren van prototype (P) konden de eindgebruikers er uitgebreid mee 'spelen'. Dit heeft geleid tot enkele aanpassingen. Nadat de eindgebruikers het prototype (P) hadden goedgekeurd (d.w.z. zij konden geen fouten meer vinden, het aangepaste prototype kon voorlopig niet gefalsificeerd worden) konden de resterende bouwactiviteiten worden verricht.
Aan de hand van de opgedane ervaringen kan worden geconcludeerd dat:

a. bij administratieve automatisering prototyping-activiteiten kunnen leiden tot een hogere betrokkenheid van de eindgebruiker/opdrachtgever;
b. de ondersteuning van de fasen definitiestudie en functioneel ontwerp met een prototype de eindgebruiker ook meer mogelijkheden biedt zijn informatiebehoeften juist en volledig te specificeren. Het specificeren zal vaak door middel van enkele 'iteratie-stappen' tot stand komen;
c. sommige eindgebruikers het zeer merkwaardig vinden dat, nadat zij een goed functionerend prototype (P) hebben zien werken, zij nog lange tijd moeten wachten voordat het informatiesysteem wordt opgeleverd;
d. het met eindgebruikers in zeer korte tijd opstellen van prototype (T) ertoe kan leiden dat het overzicht uit het oog wordt verloren. Sol [23] noemt dat het gevaar voor 'tunnelvisie';
e. ADS/ONLINE een goed hulpmiddel is om deze prototyping-activiteiten te ondersteunen.

5 INDICES

Op twee plaatsen in het programmatuursysteem 'GIRAF' spelen indices een rol, namelijk bij de toegang op naam en bij de toegang op attribuutcode + attribuutwaarde.

5.1 Toegang op naam
Gegevensgroep 1 bevat doorgaans de namen van een referentie, bijvoorbeeld de auteursnamen van een literatuurreferentie, de artikelnaam van een produkt, de naam van een persoon die lid is van een vereniging of deelneemt aan een congres, etc. 'GIRAF' biedt de mogelijkheid om aan de hand van een naam de referenties te selecteren die de betreffende naam in gegevensgroep 1 opgenomen hebben. Zo zullen bijvoorbeeld na het intoetsen van de naam 'Pietersen, K.' alle in de GTD voorkomende literatuurreferenties van de heer K. Pietersen op het beeldscherm worden getoond. Indien de naam 'Pietersen, K.' niet voorkomt in de GTD, wordt op het beeldscherm een lijst van namen getoond rondom de naam 'Pietersen, K.'. De gebruiker kan dan kijken of hij wellicht een spelfout heeft gemaakt en kan alsnog, op eenvoudige wijze, de juiste naam kiezen. Ook bij het intoetsen van een generieke naam kan een dergelijke lijst van namen worden geproduceerd. Om zo'n (alfabetisch gesorteerde) serie namen op het beeldscherm te kunnen laten zien is het noodzakelijk om alle namen van de GTD op de een of andere wijze in een gesorteerde volgorde beschikbaar te hebben. In 'GIRAF' is daartoe een SORTED SET gedefinieerd met als 'owner' het record-occurrence van recordtype GI-SYSKEN-REC (zie figuur 9c), dat de kenmerkende gegevens voor de betreffende GTD bevat, en als 'members' de record-occurrences van het recordtype GI-NAAM-REC dat alle namen, die tot de GTD behoren, bevat. Op die manier kan de gewenste serie namen worden opgespoord. Hierbij komt wel een probleem om de hoek kijken. De record-occurrences van GI-NAAM-REC zijn namelijk CALC opgeslagen. Daarmee is een bestaande naam direct benaderbaar en kan de toegang via naam op zeer snelle wijze worden gerealiseerd. Het probleem is nu dat de genoemde SORTED SET een gesorteerde CALC-TO-CALC SET is. Het doorlopen van een grote gesorteerde CALC-TO-CALC SET kost onaanvaardbaar veel tijd omdat het lezen van elk member in de set een 'echte' leesopdracht betekent. Het zoeken in een set van 10 000 namen zou gemiddeld maar liefst 5000 leesopdrachten betekenen. De geëigende oplossing in dergelijke situaties is het ma-



Figuur 13: Voorbeeld van 'ankers' in de gesorteerde set met naam-records

ken van een index. IDMS biedt mogelijkheden om te kunnen werken met indices en heeft daarvoor het zogenaamde 'Sequential Processing Facility (SPF)' [18] beschikbaar. Na bestudering van SPF waren er nogal wat vraagtekens, met name bij het optreden van veel mutaties, over de efficiëntie van dit produkt. Uit een tiental gesprekken met Amerikaanse IDMS-gebruikers die SPF reeds gebruikten, bleek weinig waardering voor SPF. Ook Cullinet erkende dat SPF verbeterd kon worden door in het najaar 1983 aan te kondigen dat een nieuwe, met IDMS geïntegreerde, index-methode werd ontwikkeld.

Zoals in hoofdstuk 1 reeds vermeld is, moesten de technische beslissingen inzake 'GIRAF' in het najaar 1983 worden genomen op basis van de op dat moment beschikbare mogelijkheden. De conclusie werd getrokken dat SPF voor 'GIRAF' niet geschikt was. Daarom is binnen 'GIRAF' een eigen index-methode toegepast. De belangrijkste uitgangspunten bij de ontwikkeling van de index-methode waren enerzijds de noodzakelijke snelheid waarmee een lijst van namen kan worden geproduceerd en anderzijds de gewenste eenvoud van de index-methode. Daarom is gekozen voor een zogenaamde anker-index die niet on-line gemuteerd wordt. In de GTD is een recordtype GI-ANKNAAM-REC opgenomen. De record-occurrences van GI-ANKNAAM-REC bevatten zogenaamde ankers. Een anker wordt afgeleid van een bestaande naam en wordt opgenomen in zowel de index als in het recordtype GI-NAAM-REC (zie figuur 9c). Er wordt voor gezorgd dat het anker altijd ongelijk is aan een naam die in de GTD voorkomt of kan voorkomen. De anker-index van een GTD wordt 's nachts met behulp van een batch-programma aangemaakt. Het aantal ankers is afhankelijk van een invoerparameter van het batch-programma dat de anker-index aanmaakt. Deze invoerparameter geeft het aantal namen aan dat tussen twee ankers voorkomt.

Indien nu een lijst van namen moet worden geproduceerd aan de hand van een generieke naam of een niet in de GTD aanwezige naam kan met behulp van de index GI-ANKNAAM-REC, die uit een moeder-index en een aantal blok-indices bestaat, het anker worden bepaald dat dicht in de buurt ligt van de naam die volgt op de ingevoerde naam. Aangezien het anker ook in de SORTED SET is opgenomen (zie figuur 13) kan via CALC dit anker direct worden bereikt en kan de gewenste serie namen worden bepaald zonder veel extra invoeropdrachten.

Samenvattend kunnen over de geschetste anker-index-methode de volgende opmerkingen worden gemaakt:

a. de ankers zijn niet bestaande namen en worden dus nooit on-line gemuteerd;
b. het muteren van de anker-index gebeurt 's nachts met behulp van een batch-programma. Daarbij wordt de oude anker-index vervangen door een nieuwe. De ankers worden afgeleid van namen die op dat moment in de GTD aanwezig zijn en zijn dus geheel afgestemd op de betreffende GTD;
c. met behulp van een parameter wordt het aantal namen tussen twee ankers bepaald. Indien voor een GTD zeer snelle responsietijden zijn vereist dan kan deze parameter bijvoorbeeld 3 of 5 zijn. Indien de responsietijden minder kritisch zijn kan een waarde 10 of groter worden genomen. Hoe kleiner de parameterwaarde hoe groter het aantal ankers;
d. tijdens het opbouwen van een GTD zal de anker-index regelmatig vernieuwd moeten worden. Het sein daartoe wordt gegeven doordat de responsietijden substantieel gaan toenemen. Desgewenst kan met behulp van een hulpprogramma de SORTED SET GI-SYSK-NAAM (zie figuur 9c) regelmatig worden geïnspecteerd. Een groot voordeel van de anker-index-methode is dat als een GTD op grootte is gebracht, de anker-index doorgaans niet meer behoeft te worden vervangen. Mutaties op de namen kunnen plaatsvinden zonder dat de anker-index behoeft te worden gewijzigd.

Geconcludeerd kan worden dat de anker-index-methode een index oplevert die zowel met betrekking tot de namen van de ankers als de (met de index samenhangende) responsietijden aangepast is aan de gebruiker, on-line niet gemuteerd wordt en doorgaans na verloop van tijd in het geheel niet meer behoeft te worden vernieuwd.

5.2 Toegang op attribuutcode + attribuutwaarde

'GIRAF' biedt mogelijkheden om on-line een verzameling referenties te selecteren met behulp van een codesysteem (zie figuur 1). Het codesysteem bestaat uit een (in principe onbeperkt) aantal attribuutcodes en per attribuutcode een attribuutwaardenverzameling. Evenals dat bij de namen het geval is, is per attribuutcode + attribuutwaarde vastgelegd welke referenties aan deze combinatie voldoen en is het recordtype GI-ATTRWD-REC met CALC (attribuutcode + attribuutwaarde) direct benaderbaar (zie figuur 9c). Selecteren op basis van een attribuutcode en een attribuutwaardeninterval is eveneens mogelijk. Ook hier ontstaat er een probleem als de grenzen van het attribuutwaardeninterval geen binnen de GTD bekende attribuutwaarden zijn. Om in dergelijke situaties toch snel de gewenste verzameling referenties te kunnen samenstellen is eveneens een anker-index geconstrueerd. Hoewel de constructie iets ingewikkelder is dan de anker-index voor namen zijn de uitgangspunten alsmede de voordelen en nadelen (ruimte voor de ankers) dezelfde.

6 CONCLUSIES

Uit de ervaringen met het ontwikkeltool ADS/ONLINE kunnen de volgende conclusies worden getrokken:

- Door de dialoogstructuur kan met ADS/ONLINE in het technisch ontwerp op een hoger abstractieniveau worden gewerkt dan met TP-monitoren zoals IDMS/DC en CICS/VS.
- De DialoogCommunicatieDiagrammen (DCD's) sluiten nauw aan bij de SchermbeeldCommunicatieDiagrammen (SCD's). Daardoor zal de overgang van functioneel ontwerp naar technisch ontwerp soepeler kunnen verlopen. Bovendien kunnen opdrachtgevers, die het functioneel ontwerp kennen, de vorderingen van het technisch ontwerp makkelijker volgen.
- Het ontwikkelen van programmatuursystemen met ADS/ONLINE zal de fase technisch ontwerp/programmering substantieel verkorten. Kwantitatieve gegevens zijn nog moeilijk te geven.
- De mogelijkheden van ADS/ONLINE, gecombineerd met een cassettesysteem zoals binnen 'GIRAF' is ontwikkeld, maken het mogelijk om ruim aandacht te besteden aan de flexibiliteitsaspecten 'open-ended' en 'adaptability'. Het leveren van maatconfectiesystemen behoort tot de mogelijkheden.
- Het werken met multiple databases kan binnen ADS/ONLINE op een soepele manier worden geregeld. Daarmee kunnen gebruikersdatabases fysiek worden gescheiden.
- ADS/ONLINE is een goed hulpmiddel om prototyping-acitiviteiten te ondersteunen. Zowel in de fasen definitiestudie, functioneel ontwerp als technisch ontwerp kunnen prototypes (T) en prototypes (P) een rol spelen. Dit kan leiden tot een hogere betrokkenheid van de eindgebruiker/opdrachtgever. Deze is bovendien beter in staat om zijn informatiebehoeften juist en volledig te specificeren. Systeemontwerpers moeten bedacht zijn op het gevaar voor 'tunnelvisie'.
- De gebruiksvriendelijkheid van het produkt On Line Mapping moet worden verbeterd.
- Hoewel Cullinet Software ernaar streeft om alle produkten te integreren zodat ze als het ware één geheel vormen, worden systeemontwerpers/programmeurs geconfronteerd met produkten die op gebruiksniveau niet altijd volledig op elkaar zijn afgestemd.

7 LITERATUUR/REFERENTIES

1. Application Development System / Online Summary Description. Cullinet Software, 1983
2. Application Development System / Online Application Design Guide, release 1.1. Cullinet Software, augustus 1983
3. Application Development System / Online User's Guide, release 1.1. Cullinet Software, augustus 1983
4. Application Development System / Online Reference Guide, release 1.1. Cullinet Software, september 1983
5. Beek, J. van; Een systeemontwikkelingsmethode gebaseerd op prototyping. In: *Informatie*, jaargang 24 nr. 12, december 1982, pp. 702-710
6. Bergstra, J. A. en J. J. Smeets; Formele ontwikkeling van schermverloopschema's en interactieve programma's. In: *Informatie*, jaargang 26 nr. 4, april 1984, pp. 318-324
7. Brussaard, B. K.; Collegedictaat 'Informatiesystemen'. Vakgroep Informatica. Technische Hogeschool Delft
8. Chapin, Ned; New format for flowcharts. In: *Software-practice and experience*, vol. 4, 1974, pp. 341-357
9. Craenen, G.; Praten met de computer. Software voor eindgebruikers. Aula, 727, Spectrum, 1983
10. Database task group (DBTG). CODASYL report april 71
11. Dijk, A. J. van; 'BIBLIOSYSTEM'. Computerhulp bij het samenstellen van bibliografieën. In: *Informatie*, jaargang 13 nr. 10, oktober 1971, pp. 448-455
12. Dijk, A. J. van; 'BIBINFO'; Een real-time retrieval systeem. Afstudeerverslag, vakgroep Informatica TH-Delft, Memorandum no. RC-DMS-76002, november 1976, 236 pag.
13. Dijk, A. J. van; Ervaringen met het COMMAND-level van CICS/VS bij de ontwikkeling van een geautomatiseerde bibliotheek-toepassing. In: *Informatie*, jaargang 22 nr. 4, april 1980, pp. 313-322
14. Dijk, A. J. van; 'GIRAF'; Een information retrieval systeem voor universiteiten, hogescholen en andere pluriforme organisaties. In: *Informatie*, jaargang 26 nr. 6, mei 1984, pp. 445-455
15. Dijk, A. J. van en R. de Roos; 'GIRAF' – General Information Retrieval Facilities. Functioneel ontwerp. Technische Hogeschool Delft, Rekencentrum. Memorandum no. RC-ISO-82012, december 1982, 213 pag.
16. Dijk, A. J. van, R. P. Karlas, P. Makkes en R. J. J. van Schie; 'GEMS' – Geautomatiseerd Magazijnadministratie- en voorraadbeheerSysteem. Functioneel ontwerp. Technische Hogeschool Delft, Rekencentrum Memorandum no. RC-ISO-84004, juni 1984, 256 pag.
17. Eilers, H. B.; Systeemontwikkeling volgens SDM, Academic Service Den Haag, 1980, ISBN 90 623 3043 6
18. IDMS, Sequential Processing Facility. Cullinet Software, april 1983
19. IDMS-DC, Concepts and Facilities. Cullinet Software, april 1979
20. Janssens, H.; Gestruktureerde conversaties. In: *Informatie*, jaargang 24 nr. 3, maart 1982, pp. 152-160
21. Ooninex, J. A. M.; Waarom falen informatiesystemen nog steeds? Samsom, Alphen a/d Rijn, 1982, ISBN 90 140 3201 3
22. Rees, J. R. van; Ontwikkelingsmethoden 22. De methode doet het niet. In: *Informatie*, jaargang 24 nr. 2, februari 1982, pp. 81-93
23. Sol, H. G.; Ervaringen rond prototyping. In: *Informatie*, jaargang 26 nr. 3, maart 1984, pp. 236-242

24. De auteur van dit artikel is dank verschuldigd aan de collega's R. de Roos en R. J. J. van Schie voor de vele interessante discussies die hebben bijgedragen aan de totstandkoming van dit artikel.

Publication 11

BIBLIOTHEEK

Delftse studenten vragen computer

om boeken uit de bibliotheek

Students Delft University of Technology ask computer for library books

1979

The decision to pressure people into delivering a product that doesn't measure up to their own quality standards is almost always a mistake.

Alan Davis [1993]

This article previously appeared in:

DP-Monitor, April 1979, pages 16-23

Abstract
The Library of the Delft University of Technology (DUT) has received a lot of international attention since 1963 because of its unique system for requesting books: amongst other things using the 'bibliofoon'. In 1977, the Library heard of an integrated library system that had been developed in Dortmund: DOBIS (Dortmund Bibliotheks-system). This article discusses the possibilities of DOBIS for the DUT and describes an intermediary request system that was designed, built and introduced by the DUT because DOBIS in 1977 did not yet include a request system that was suitable for the readers.

Ir. A. J. van Dijk,
medewerker afdeling
Ontwikkeling
Informatiesystemen,
Rekencentrum
TH-Delft

Delftse studenten vragen computer om boeken uit de bibliotheek

Reeds in de 7de eeuw werden in China boeken gedrukt, zogenaamde 'blokboeken'. Het drukken met losse letters werd daar in 1041 uitgevonden. In West-Europa waren blokboeken eerst in het begin van de 15de eeuw bekend. Hier werd het drukken met losse letters uitgevonden omstreeks 1450 en de uitvinding werd toegeschreven aan Laurens Jansz. Coster te Haarlem of aan Johannes Gutenberg te Mainz.

Sinds de beginjaren van het boekdrukken hebben mensen zich beziggehouden met het verzamelen van boeken. Al gauw ontstonden op deze manier veel particuliere boekverzamelingen. Sommige van deze boekverzamelingen werden opengesteld voor meer algemeen gebruik, waarmee de eerste particuliere bibliotheken ontstonden. In Nederland kent men naast particuliere bibliotheken : wetenschappelijke bibliotheken, openbare leeszalen en bibliotheken, bedrijfsbibliotheken en leesbibliotheken.

De bibliotheek van de Technische Hogeschool Delft bestaat sinds 1842 en is een wetenschappelijke bibliotheek, die naast de functie van bibliotheek voor de TH-Delft ook de landelijke functie van Centrale Technische Bibliotheek vervult. De collectie, hoe uitgebreid ook (± 554.000 delen, waarvan 332.000 boeken en 222.000 tijdschriftbanden), omvat uiteraard slechts een gedeelte van de wereldliteratuur op technisch en aanverwant terrein. Om de gebruiker van de bibliotheek zoveel mogelijk behulpzaam te zijn, is een nauwe samenwerking met andere bibliotheken in binnen- en buitenland opgebouwd : door middel van telex en documentreproduktie kan de benodigde literatuur, wanneer deze in Delft niet aanwezig is, van elders worden betrokken.

De Delftse bibliotheek staat sinds 1963 in de internationale belangstelling vanwege de unieke wijze van het aanvragen van boeken : onder andere met behulp van de 'bibliofoon' (1). De bibliofoon bracht een directe verbinding tot stand tussen de bezoeker en het boekenmagazijn. Met behulp van de kiesschijf van een telefoontoestel draaide de bezoeker het plaatsnummer van het boek. Een combinatie van lampen, die in het boekenmagazijn oplichtten, vertelde de magazijnbediende waar het boek stond dat gevraagd werd. Het boek werd daarop naar het uitleenbureau getransporteerd en het boeknummer werd, voorzien van een + of − teken, ingetoetst. In de aanvraagruimte werden dan het boeknummer en + of − teken zichtbaar gemaakt op het bibliofooncontrolebord.

Het voordeel van de bibliofoon was, dat de aanvrager niet behoefde te wachten tot zijn verzoek in het uitleenbureau in behandeling kon worden genomen, maar dat hij snel en rechtstreeks zelf zijn opgave kon doen. Bovendien behoefde hij geen formulieren in te vullen voor boeken die niet aanwezig bleken te zijn. Een en ander betekende een belangrijke tijdbesparing, zowel voor de bezoeker als voor het uitleenbureau.

Sinds maart 1976 is de oude Delftse bibliofoon vervangen door een nieuw apparaat (2). Deze vervanging was noodzakelijk vanwege de slijtage van de oude bibliofoon, die oorzaak was van veel storing en uitval. Het nieuwe apparaat is betrouwbaarder en heeft een grotere capaciteit, maar biedt de gebruiker niet meer dan dat. Kenmerkend voor de Delftse situatie is dus, dat de gebruiker zelf zijn boeken aanvraagt via de apparatuur van het aanvraagsysteem.

In 1968 verscheen in de bibliotheek het eerste rapport over integrale automatisering. Daarna zijn er in de periode 1968–1977 nog een aantal rapporten over dit onderwerp verschenen. Hoewel de rapporten van goede kwaliteit waren, kwam de bibliotheek er toch niet toe om de uitwerking ter hand te nemen en een systeem te realiseren. Wel zijn enkele deelsystemen gerealiseerd zoals het Thesaurusproject en de automatisering van de Centrale Technische Catalogus.

In 1977 kwam het de bibliotheek ter ore dat er in Dortmund een geïntegreerd bibliotheeksysteem was ontwikkeld, dat wellicht voor Delft interessant was. Het betrof hier het Dortmund Bibliotheks-system (DOBIS). Dit systeem bleek inmiddels ook in gebruik te zijn bij de Katholieke Universiteit te Leuven (K.U.L.). De K.U.L. was bezig een aantal veranderingen en uitbreidingen aan te brengen. De K.U.L. noemde deze Leuvense versie van het Dortmundse systeem : LIBIS.

Zodra een werkgroep van de bibliotheek de bibliotheektechnische aspecten van DOBIS had bekeken, werd het Rekencentrum van de TH-Delft gevraagd te onderzoeken of het mogelijk was het DOBIS en/of LIBIS systeem op het computersysteem van het rekencentrum te implementeren. Na een preliminaire studie (3) kon deze vraag bevestigend worden beantwoord.

De centrale hal in de bibliotheek van de Technische Hogeschool Delft

In dit artikel zal eerst het systeem DOBIS/LIBIS in het kort worden belicht, daarna zal de eerste fase van de implementatie worden beschreven.

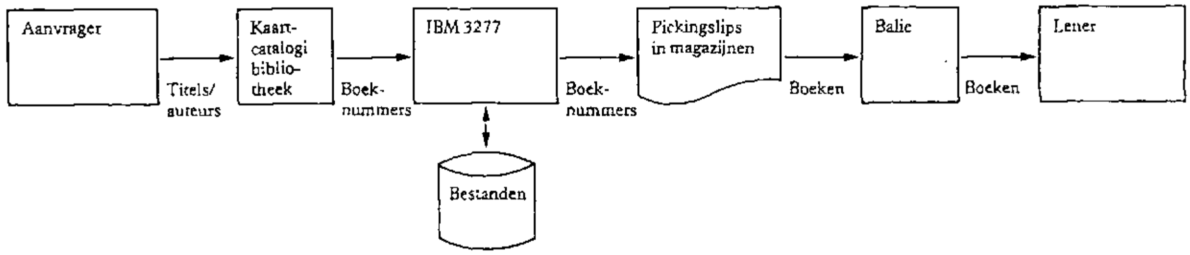
Het systeem DOBIS

Het systeem DOBIS is een produkt van de Gesamthochschule Dortmund. In deze Gesamthochschule participeren: de Universitaet Dortmund, de Paedagogische Hochschule Ruhr en de Fachhochschule.

DOBIS is gemaakt om de drie universiteitsbibliotheken, alsmede 25 afdelingsbibliotheken te automatiseren. Kenmerkend hierbij is, dat het boekenbezit van de gezamenlijke bibliotheken als één geheel wordt beschouwd, terwijl de diverse boekverzamelingen gedecentraliseerd zijn opgeslagen. DOBIS automatiseert slechts één of twee functies van het bibliotheekgebeuren, het is een compleet informatiesysteem. De hoofdfuncties van DOBIS zijn: bestellen, ontvangen, catalogiseren, administratie van inbinden en uitlenen. Enkele kenmerkende eigenschappen van DOBIS zijn:

- De communicatie, die via beeldschermterminals plaatsvindt, kan in verschillende talen verlopen waaronder de Nederlandse taal.
- Alle bestanden zijn direct toegankelijk, zodat vastlegging van gegevens snel kan geschieden en de informatie up-to-date is.
- DOBIS is gebruikersvriendelijk. Dit wordt onder andere bereikt door de mogelijkheid van het ketenen van commando's, hetgeen betekent dat een ervaren gebruiker naar believen stappen in de dialoog kan overslaan.
- De op de IBM 3270 beeldschermterminal aansluitbare streepjescodelezer kan zinvol worden gebruikt bij toepassing van DOBIS in een bibliotheek.
- De DOBIS bestanden kunnen onder andere worden geraadpleegd door middel van: auteursnaam, titel, onderwerp, plaatsnummer, uitgever, ISBN en ISSN, projectnummer, contractnummer, lener en leverancier.

De bibliotheek van Dortmund is een 'open bibliotheek'. Dit betekent dat de aanvrager zelf zijn boeken uit de rekken haalt en zich met deze boeken aan de uitleenbalie vervoegt. Het bibliotheekpersoneel verzorgt de uitlening. De aanvrager behoeft dus niet zelf het beeldschermterminal IBM 3270 te bedienen. Ter verhoging van de produktiviteit en de betrouwbaarheid zijn



de beeldschermterminals voorzien van een streepjescodelezer. Zowel lenerskaart als boeken zijn voorzien van een streepjescode.

In Delft ligt de situatie duidelijk anders, omdat de bibliotheek van het gesloten type is. De (potentiële) aanvrager zoekt in de catalogi naar de door hem gewenste boeken en noteert de boeknummers. Zoals in de inleiding is vermeld, zal hij daarna zelf de boeknummers van de door hem gewenste boeken intoetsen en wachten totdat zijn aanvraag is afgehandeld. Dit stukje dialoog is niet aanwezig in DOBIS en zal door de Technische Hogeschool Delft zelf moeten worden ontwikkeld. Belangrijk daarbij is, dat dit stukje van de dialoog zeer gebruikers-vriendelijk moet zijn, omdat van een bezoeker van de bibliotheek niet mag worden verwacht, dat hij een ervaren beeldschermterminal-gebruiker is.

Een kenmerkend verschil van het Leuvense systeem LIBIS met DOBIS is, dat LIBIS algemener is met betrekking tot formaatstructuren. Verder is LIBIS gebaseerd op een netwerkstructuur. Dit betekent dat afdelingsbibliotheken als het ware kunnen aanhaken bij het LIBIS systeem van de (centrale) bibliotheek, zonder daarbij hun eigen identiteit of regels over boord te gooien. Aangezien men aan de TH-Delft te zijner tijd de afdelingsbibliotheken eveneens wil interesseren voor DOBIS, is dit laatste zeer belangrijk. Dit was dan ook één van de belangrijkste redenen voor de TH-Delft om te kiezen voor LIBIS in plaats van DOBIS. Een andere reden was de Leuvense uitleenfunctie, die voor Delft interessanter was dan de uitleenfunctie van DOBIS.

Het invoeren van een systeem als LIBIS kan niet van de ene op de andere dag geschieden. Daartoe dient een invoeringsplan gemaakt te worden. De TH-Delft heeft zo'n invoeringsplan gemaakt en een onderdeel van dat invoeringsplan is de fasering. Alvorens deze fasering te beschrijven, moet nog worden vermeld dat, op het moment dat de beslissing door de bibliotheek van de TH-Delft over aanschaf moest worden genomen, het systeem LIBIS nog niet geheel gereed was. De TH-Delft heeft toen besloten om als tussenstap naar LIBIS zelf een intermediair aanvraagsysteem te ontwikkelen.

De voordelen van het intermediair aanvraag-systeem zijn:

a. De aanvrager krijgt binnen enkele seconden te zien of de gevraagde boeken al dan niet aanwezig zijn.

b. Boeknummers van boeken die niet aanwezig zijn worden niet doorgegeven aan het boeken-magazijn. Momenteel worden alle boeknummers van aangevraagde boeken doorgestuurd naar het boekenmagazijn. Aangezien ± 40% van de aangevraagde boeken uitgeleend is, betekent het 'afvangen' van de aanvragen van boeken die uitgeleend zijn een forse werkbesparing voor het magazijnpersoneel, dat de vrijkomende tijd aan ander belangrijk werk in het boekenmagazijn kan besteden.

c. De opbouw van de LIBIS bestanden kan geleidelijk aan plaatsvinden. Het ligt in de bedoeling om de LIBIS bestanden op te bouwen door de boeken die zijn uitgeleend, bij terugbezorging daarin op te nemen. Op die manier worden deze bestanden gevuld met het meest 'actieve' deel van het boekenbezit. Wanneer nu de LIBIS bestanden voldoende omvang hebben en de betreffende functies van het (aangepaste) systeem LIBIS geïmplementeerd zijn, kan worden overgeschakeld van het inter-mediaire aanvraagsysteem naar de LIBIS uitleen-functie.

d. Klanten en personeel van de bibliotheek kunnen alvast wennen aan het IBM 3270 beeld-schermterminal.

e. De programmatuur van LIBIS is gebaseerd op het programmatuursysteem CICS/VS van IBM. Het Customer Information Control System (CICS) is een berichtenbesturingssysteem, ontworpen voor het behandelen van transacties in een data base/data communicatie omgeving. Door nu de programmatuur van het intermediair aanvraag-systeem eveneens te baseren op CICS/VS kan de projectgroep ervaring opdoen in het werken met CICS/VS.

f. Het bibliotheekpersoneel wordt op korte termijn met daadwerkelijke resultaten geconfronteerd en de verdere invoering van LIBIS kan stap voor stap plaatsvinden. Deze stap voor stap methode is zeer belangrijk, omdat het bij LIBIS gaat om een compleet informatiesysteem. Door stap voor stap, functie voor functie te implementeren, kan de projectgroep de betreffende afdeling van de bibliotheek de nood-zakelijke begeleiding geven.

g. Statistische gegevens komen snel beschikbaar.

h. Er behoefde niet te worden gewacht op realisatie van de nog in bewerking zijnde functies van LIBIS.

Om de invoering volgens bovengenoemde stap voor stap methode te laten plaatsvinden, is de volgende fasering aangebracht:
fase 1: aanvraagprocedure via intermediair systeem;
fase 2: aanvraag- en uitleenprocedure LIBIS;
fase 3: statistische bedrijfsgegevens (vaak als onderdeel van andere fasen);
fase 4: beheer van tijdschriften en seriewerken;
fase 5: catalogisering en aanschaf;
fase 6: diensten ten behoeve van andere TH-bibliotheken;
fase 7: mechanisch magazijn voor de actieve collectie.

In februari 1979 is de programmatuur voor fase 1 opgeleverd. De bibliotheek is toen gestart met een proefperiode die duurt tot en met mei. In deze periode kunnen bezoekers en personeel van de bibliotheek kennis maken met het intermediair aanvraagsysteem. De oude uitleenprocedure blijft in deze periode gehandhaafd. Het intermediair aanvraagsysteem werkt dan met schaduwbestanden. De projectgroep werkt in deze maanden aan een noodprocedure die moet gaan functioneren als het computersysteem uitvalt. Bovendien worden alle procedures van administratief-organisatorische aard rondom het intermediair aanvraagsysteem nauwkeurig vastgelegd.

Met de invoering van het intermediair aanvraagsysteem omstreeks mei 1979 zal de integrale automatisering van de bibliotheek TH-Delft zijn gestart.

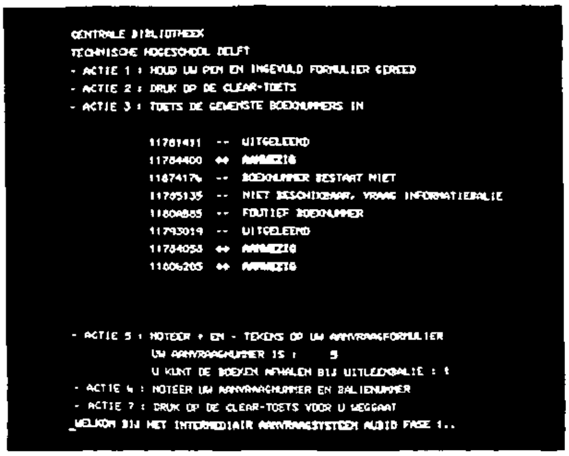
Het intermediaire aanvraagsysteem

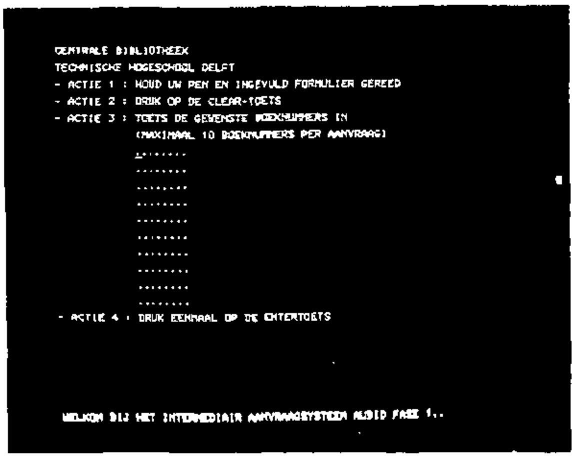
Voordat met de bouw van het intermediair aanvraagsysteem werd begonnen, zijn een aantal wensen geformuleerd waaraan het systeem moest voldoen. De volgende functies moesten in het systeem aanwezig zijn.

- Het genereren van een statusbestand, dat de boekstatus van ieder boek bevat. Zo'n boekstatus kan zijn: aanwezig, uitgeleend, niet beschikbaar, enzovoort.
- Van boeken met duplicaten het aantal exemplaren vastleggen.
- Boekaanvragen verwerken en aan de hand van de aanvragen het statusbestand bijwerken.
- Van de aangevraagde boeken (ook voor duplicaten) die aanwezig blijken te zijn, het boeknummer en eventuele andere relevante

In de proefperiode die deze maand zal worden afgesloten hebben bezoekers en personeel van de bibliotheek al de nodige ervaring met het intermediair aanvraagsysteem op kunnen doen

De hier afgedrukte serie foto's (die doorloopt op pag. 21) geeft een goed beeld van de wijze waarop de boekaanvrager door het systeem wordt begeleid. De benodigde programmatuur werd grotendeels ontwikkeld door medewerkers van de afdeling Ontwikkeling Informatiesystemen (3e foto van links)





informatie afdrukken op de matrixprinter van het betreffende boekenmagazijn.

- Terugbezorgingen verwerken, hetgeen onder andere betekent dat het statusbestand moet worden bijgewerkt.
- Nieuwe boeknummers (ook van duplicaten) in het statusbestand opnemen.
- Een statistiekbestand met betrekking tot de gepleegde aanvragen opbouwen en overzichten produceren.

De bovenstaande wensen werden aangevuld met wensen die naar voren kwamen uit de interviews met bibliotheekmedewerkers.

Allereerst wordt nu de apparatuur besproken die in de bibliotheek staat opgesteld; vervolgens zal het systeemontwerp aan de orde komen.

Apparatuur

De apparatuur bestaat uit IBM 3277 beeldschermterminals en matrixprinters van het type IBM 3287. In de aanvraaghal staan vier beeldschermterminals. Daarnaast heeft de afdeling 'Uitleen' twee beeldschermterminals, één voor de uitleenbalie en één voor de terugbezorgbalie. De afdeling 'Informatie', die midden in de aanvraaghal is gesitueerd, heeft één beeldschermterminal evenals de afdeling 'PTT', die de aanvragen die via de Post, de Telefoon en de Telex binnenkomen afhandelt.

De vier boekenmagazijnen beschikken elk over een matrixprinter. Ten behoeve van de verdere ontwikkeligen rondom LIBIS zijn nog twee beeldschermterminals alsmede een lijnprinter met bar-code-feature opgesteld. De laatste twee beeldschermterminals fungeren tevens als reserve terminals, en kunnen worden ingezet als één van de acht andere beeldschermterminals het plotseling begeeft.

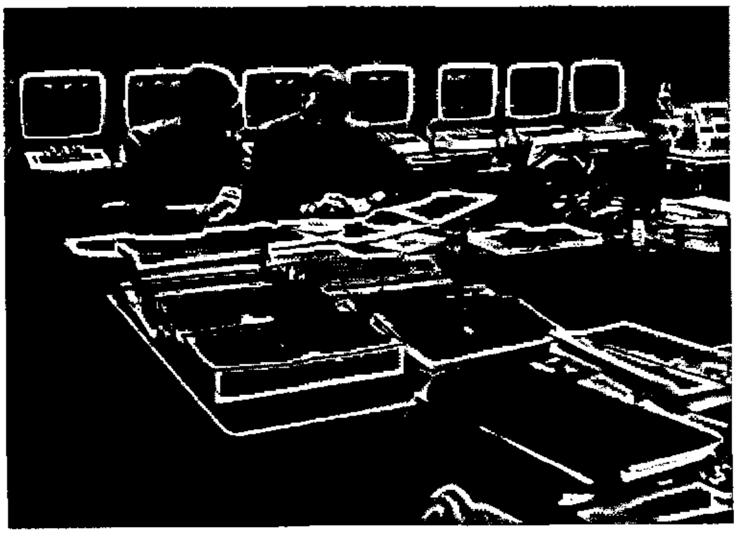
Programmatuur

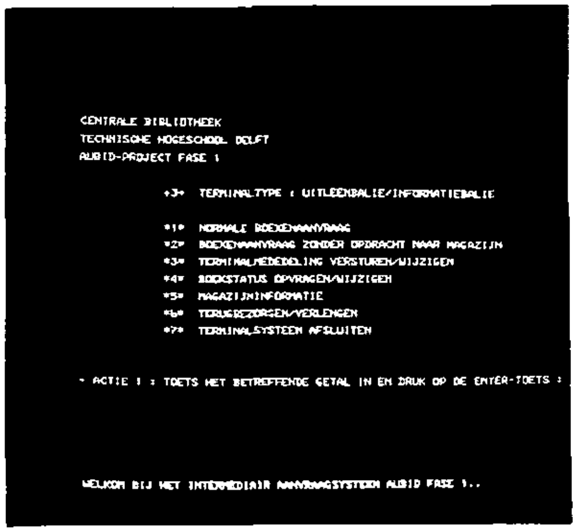
In het functioneel ontwerp is het intermediair aanvraagsysteem als het ware in vier stukken verdeeld. Deze vier stukken zijn de basisfuncties van het intermediair aanvraagsysteem. Iedere beeldschermterminal kan slechts één van deze vier functies tegelijk 'bedienen'. Bij het opstarten krijgt het bibliotheekpersoneel de gelegenheid om aan iedere beeldschermterminal de gewenste functie toe te kennen. De vier functies zijn:
functie 1: terminal voor halaanvragen (Nederlands);
functie 2: terminal voor halaanvragen (Engels);
functie 3: terminal voor informatiebalie of uitleenbalie;
functie 4: terminal voor PTT-aanvragen.

Het is niet zonder meer mogelijk om van de ene functie in de andere functie te komen. Wel heeft het bibliotheekpersoneel de mogelijkheid om via een wachtwoord een functie te verlaten en opnieuw een functie aan het onderhavige beeldschermterminal toe te kennen. Deze werkwijze heeft twee voordelen. In de eerste plaats is er een duidelijke functiescheiding bereikt, de aanvragers in de hal kunnen de bibliotheekfuncties niet bereiken. In de tweede plaats is het systeem flexibel met betrekking tot het gebruik. Normaliter zullen de vier beeldschermterminals in de hal van functietype 1 zijn. Wanneer nu een groep buitenlanders de bibliotheek bezoekt (bij voorbeeld cursisten van de internationale cursussen die aan de TH worden gegeven), kan het bibliotheekpersoneel één of meer beeldschermterminals in de aanvraaghal veranderen in een functietype 2 terminal, waardoor de dialoog in de Engelse taal verloopt. Ook kan het bibliotheekpersoneel tijdelijk gebruik maken van een beeldschermterminal in de aanvraaghal door deze te veranderen in een functietype 3 terminal. Dit laatste kan nodig zijn als plotseling één van de beeldschermterminals van het bibliotheekpersoneel uitvalt, of als er erg veel werk aan de terminals gedaan moet worden en het in de aanvraaghal erg rustig is.

Functietypes

De bezoeker van de bibliotheek die, na de catalogi te hebben geraadpleegd, een





formuliertje met de gewenste boeknummers heeft ingevuld, wordt vervolgens geconfronteerd met de beeldschermterminal en de op het scherm van deze terminal vermelde gegevens. Naast het beeldschermterminal staat een kaart met enkele simpele gebruiksaanwijzingen. De aanvrager moet nu de acties uitvoeren die op het scherm vermeld staan, te beginnen bij actie 1.

Nadat de gewenste boeknummers zijn ingetoetst, wordt middels actie 4 op de enter-toets gedrukt en wordt de aanvraag in behandeling genomen. Mocht de aanvrager handelingen verrichten die hij niet mag verrichten, bij voorbeeld op toetsen drukken die voor het aanvragen van boeken niet relevant zijn, dan verschijnt er een mededeling op regel 1 van het scherm. De lichtintensiteit van deze mededeling is hoger dan die van de andere gegevens op het scherm, zodat de mededeling goed opvalt. De aanvrager behoeft zich dus geen zorgen te maken dat hij iets onherstelbaar fout zou doen.

Naast de boeknummers van boeken die aanwezig zijn, en die nu de status uitgeleend bezitten, wordt met hoge lichtintensiteit '+ +' op het scherm gezet. De niet gehonoreerde boeknummers worden voorzien van de tekens '— —' met normale lichtintensiteit. Achter de tekens '+ +' verschijnt doorgaans de tekst 'aanwezig'. Soms verschijnt er een andere tekst die betrekking heeft op het magazijn dat het gevraagde boek bezit, bij voorbeeld 'boek wordt elders opgehaald, duur ca. 30 minuten'. De bibliotheek kan namelijk bij ieder magazijn een speciale boodschap definiëren die in plaats van de tekst 'aanwezig' op het scherm verschijnt. Achter de tekens '— —' verschijnt de reden van het niet kunnen voldoen aan de aanvraag, bij voorbeeld 'uitgeleend' of 'foutief boeknummer'.

Iedere aanvrager die tenminste eenmaal '+ +' heeft gekregen, krijgt een aanvraagnummer. De gehonoreerde boeknummers worden op zogenaamde 'pickingslips' doorgestuurd naar de betreffende magazijnen. Ieder boeknummer staat op een afzonderlijke pickingslip. Behalve het boeknummer staat op deze pickingslip onder andere het aanvraagnummer en het aantal boeken dat de aanvrager zal ontvangen. Dit vergemakkelijkt het werk van het personeel van de uitleenbalie, dat de van de lopende band afkomende boeken direct op aanvraagnummer kan leggen en bovendien kan constateren of alle boeken voor een aanvrager al bij de uitleenbalie gearriveerd zijn. In het laatste geval kan het personeel desgewenst via de microfoon de betreffende aanvrager waarschuwen dat zijn boeken gereed liggen.

Naast het aanvraagnummer verschijnt ook het balienummer op het scherm. Momenteel is er slechts één uitleenbalie, maar op verzoek van de bibliotheek is er in het intermediair aanvraagsysteem rekening gehouden met meer balies. Een tweede uitleenbalie wordt wellicht in de nabije toekomst geïnstalleerd.

Nadat de aanvrager de acties 6 en 7 heeft uitgevoerd, kan hij óf met een nieuwe aanvraag starten, óf het beeldschermterminal vrijgeven voor een volgende aanvrager.

Het aanvragen van boeken via een functietype 2 terminal verloopt hetzelfde als het aanvragen van boeken via een functietype 1 terminal. Alle door de programmatuur gegenereerde teksten zijn echter in de Engelse taal gesteld.

Het bibliotheekpersoneel beschikt over een functie die een aantal mogelijkheden biedt om het bedrijfsgebeuren binnen de servicediensten soepel te laten verlopen. De afdeling 'Uitleen' en de afdeling 'Informatie' beschikken over beeldschermterminals waarop normaliter functietype 3 functioneert. De afdeling 'PTT' heeft op haar beeldschermterminal de beschikking over functietype 4.

Functietype 3 omvat de volgende subfuncties:
- normale boekenaanvraag;
- boekenaanvraag zonder opdracht naar magazijn;

- terminalmededeling versturen/wijzigen;
- boekstatus opvragen/wijzigen;
- magazijninformatie;
- terugbezorgen/verlengen;
- terminalsysteem afsluiten.

Subfuncties
Met de subfunctie normale boekenaanvraag kan het personeel van de servicediensten zelf boeken aanvragen. In plaats van een aanvraagnummer komt op de pickingslip het woord 'balie' te staan. De subfunctie kan bij voorbeeld worden gebruikt om bezoekers die aan de balie komen te helpen met het aanvragen van een boek.

Soms komt het voor dat een bezoeker van de bibliotheek bij de afdeling 'Uitleen' een boek ziet liggen, dat terugbezorgd is en dat hij graag wil lenen. Het is dan niet zinvol om het boek eerst naar het magazijn te brengen, via een beeldschermterminal aan te vragen en weer uit het magazijn te halen. In deze situatie kan een boek worden aangevraagd, zonder dat er een pickingslip naar het magazijn wordt verstuurd.

De subfunctie terminalmededeling versturen/wijzigen kan gezien worden als een 'intercom' mogelijkheid. Vanaf een functietype 3 terminal kan een boodschap worden verstuurd naar alle terminals van functietype 1, naar alle terminals van functietype 2, naar alle terminals van functietype 3, naar alle terminals van functietype 4 of naar alle beeldschermterminals. De boodschap wordt op regel 24 van de betreffende beeldschermterminals zichtbaar gemaakt met een hoge lichtintensiteit. Het personeel van de bibliotheek heeft de mogelijkheid om een tiental mededelingen die regelmatig kunnen voorkomen in het systeem te laten opnemen.

Zodra een mededeling moet worden verstuurd, verschijnen deze vastgelegde mededelingen op het scherm, waarna men slechts een getal tussen 1 en 10 behoeft in te toetsen. Eenmalige mededelingen kunnen vanzelfsprekend ook worden verstuurd. De vastgelegde mededelingen kunnen op ieder gewenst moment worden gewijzigd. Verstuurde mededelingen zullen onmiddellijk op de betreffende schermen verschijnen en na iedere actie vanachter die beeldschermterminals opnieuw verschijnen. Alleen door het sturen van een blanco mededeling kan de oude mededeling worden verwijderd.

De leeszaal van de bibliotheek

Boekstatus opvragen/wijzigen

In iedere werkomgeving worden fouten gemaakt, dus ook bij de servicediensten van de bibliotheek. Wanneer blijkt dat een boek een verkeerde status heeft, kan de status worden gewijzigd met deze subfunctie. Wanneer een boek tijdelijk niet kan worden uitgeleend, omdat het bij voorbeeld naar de binderij is of omdat het 'zoek' is, kan de boekstatus worden gezet op: 'niet beschikbaar'. Met deze subfunctie is het tevens mogelijk nieuwe boeknummers in te voeren.

Magazijninformatie

Soms is het gewenst een boodschap te sturen naar een boekenmagazijn, bij voorbeeld als er een verkeerd boek naar de uitleenbalie getransporteerd is en het personeel van de uitleenbalie alsnog het goede boek wil hebben. De subfunctie magazijninformatie biedt de mogelijkheid om een boodschap naar één of meer magazijnen te sturen. Bij de beschrijving van functietype 1 is de mogelijkheid genoemd om per magazijn een speciale boodschap te definiëren die, bij het aanwezig zijn van aangevraagde boeken, op het scherm verschijnt achter de tekens '+ +'. In deze subfunctie kan per magazijn een dergelijke speciale boodschap worden gedefinieerd.

Normaliter is aan boekenmagazijn 1 matrixprinter 1 toegewezen, boekenmagazijn 2 ontvangt de aanvragen en boodschappen via matrixprinter 2, enzovoort. Soms is het gewenst een boekenmagazijn aan een andere matrixprinter te koppelen. Bij voorbeeld als een matrixprinter kapot is, kunnen de aanvragen tijdelijk via een andere matrixprinter worden verstuurd. Ook uit bedrijfsorganisatorisch oogpunt is het soms wenselijk meerdere magazijnen te verbinden met één matrixprinter, bij voorbeeld wanneer in stille tijden één magazijnmedewerker meerdere magazijnen bedient. Met de subfunctie magazijninformatie kan op ieder moment een willekeurige matrixprinter aan een magazijn worden gekoppeld.

Terugbezorgen/verlengen

Wanneer de uitgeleende boeken worden terugbezorgd, moet de boekstatus van de betreffende boeken op 'aanwezig' worden gezet. Daartoe moeten de boeknummers worden ingetoetst. Wanneer een uitlening moet worden verlengd, wordt door het intermediair aanvraagsysteem de eerste uitlening beëindigd en een tweede uitlening gestart.

Aan het eind van de dag, in de bibliotheek doorgaans om 21.30 uur, kan het terminalsysteem worden afgesloten. Na het activeren van deze subfunctie zal het intermediair aanvraagsysteem nog een aantal handelingen verrichten, zoals het afsluiten van het statistiekbestand.

Statische gegevens

Het intermediaire aanvraagsysteem verzamelt een aantal statistische gegevens, die betrekking hebben op de ingetoetste boeknummers alsmede op het gebruik van het systeem. Van ieder ingetoetst boeknummer wordt een statistiekrecord gemaakt, dat naast boeknummer het tijdstip van aanvraag alsmede een code bevat. De code geeft aan wat er met het boeknummer aan de hand was, bij voorbeeld aanwezig, uitgeleend, foutief boeknummer. In totaal bestaan er 31 verschillende codes.

Met betrekking tot het gebruik van het intermediair systeem worden onder andere vastgelegd:

- het indrukken van de clear-toets;
- het indrukken van de enter-toets;
- het indrukken van een verkeerde toets;
- het activeren van een subfunctie van functietype 3 en het wisselen van functietype.

Het dagelijkse statistiekbestand wordt toegevoegd aan een historisch statistiekbestand. Een statistiekprogramma produceert overzichten per code, per dagnummerinterval en per tijdsinterval.

Referenties

(1) J. Verhoeff: *The Delft Circulation System*. Libri 1966, vol. 16, no. 1, pp. 1–9.

(2) Bibliotheek TH-Delft: *Bibliotheekberichten*. TH-Delft, januari 1978.

(3) A. J. van Dijk, M. Looijen: *Verslag van de preliminaire studie van het systeem* DOBIS/LIBIS *in relatie tot de Centrale Bibliotheek*. Rekencentrum TH-Delft, memorandum RC/ISO/77001 Delft, april 1977.

(4) IBM: *Dortmund Library System*-DOBIS. Executive Guide, Guide for Application Programming No. GE15-6068-0, New York, 1975.

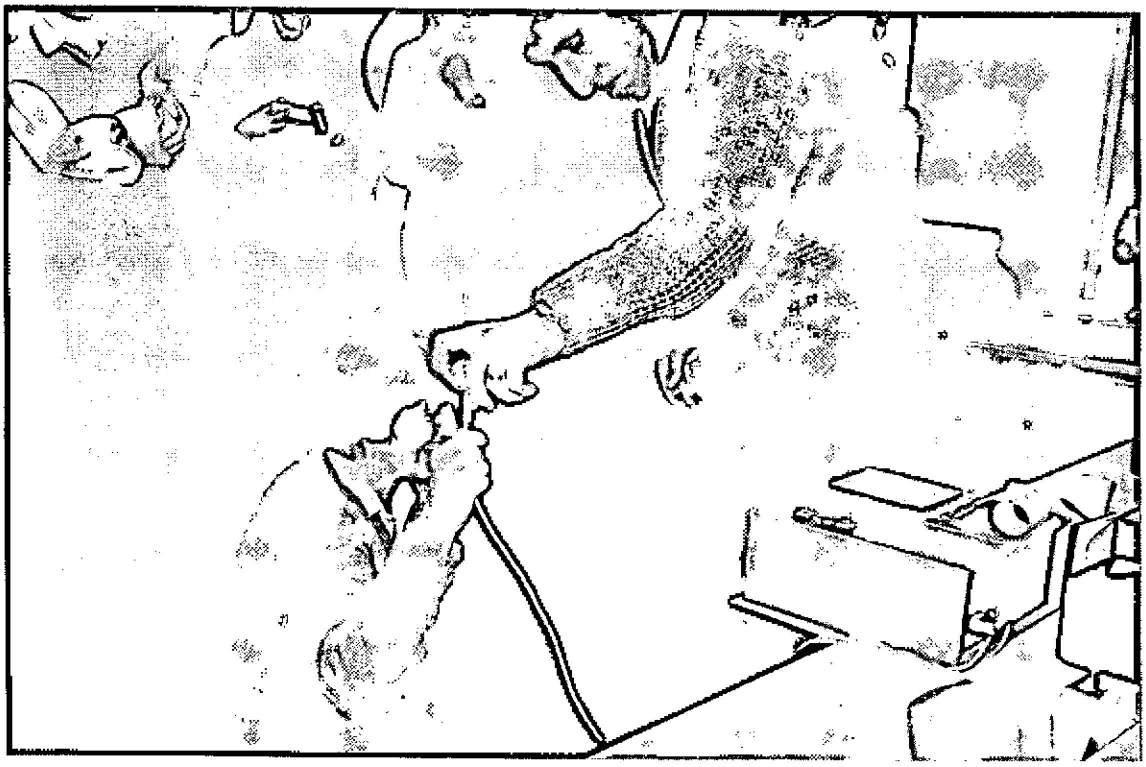
(5) IBM: *Dortmund Library System*-DOBIS. Application Guide, Guide for Application Programming No. GE15-6069-0, New York, 1975.

(6) A. J. van Dijk, J. C. Kwakernaak: AUBID-*project, Fase 1, Opbouwen van het uitleenbestand*. Delft, februari 1978, 123 pag.

(7) A. J. van Dijk, J. C. Kwakernaak: AUBID-*project, Fase 1, Systeemontwerp en programmabeschrijving*. Delft, april 1979, 150 pag.

31 oktober 1979
Ingebruikneming van het nieuwe systeem

De laatste werkzaamheden worden verricht

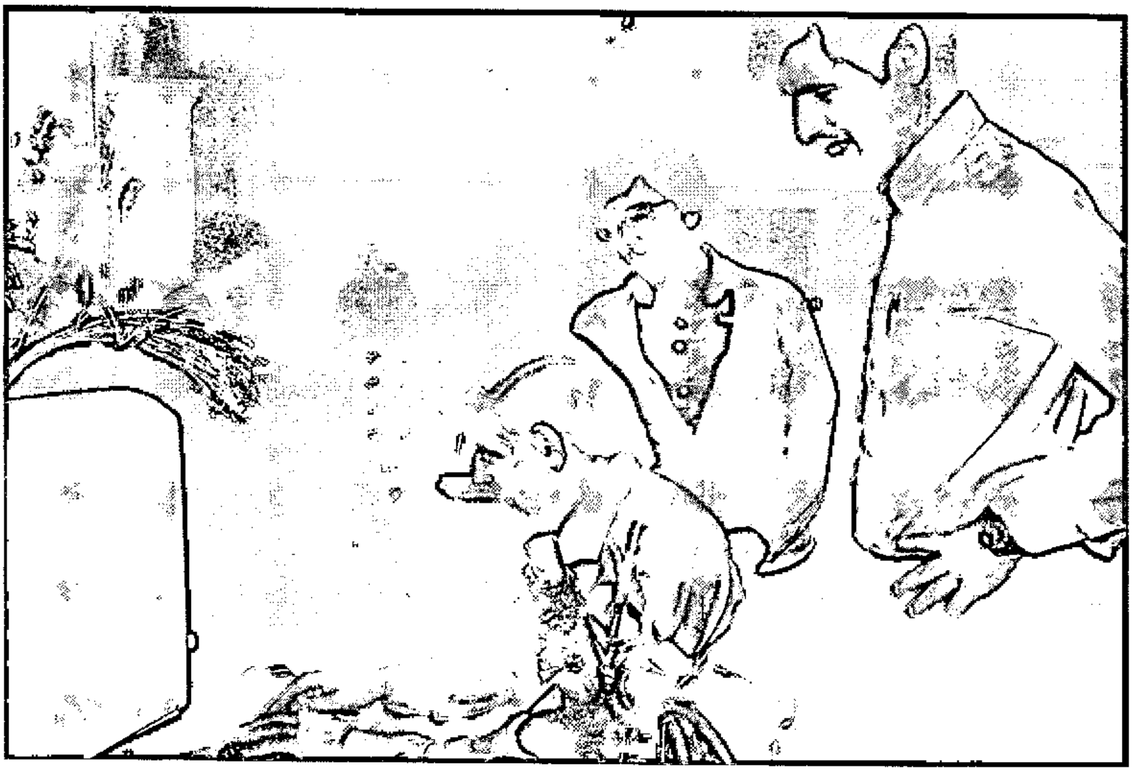


Het oude systeem wordt afgesloten

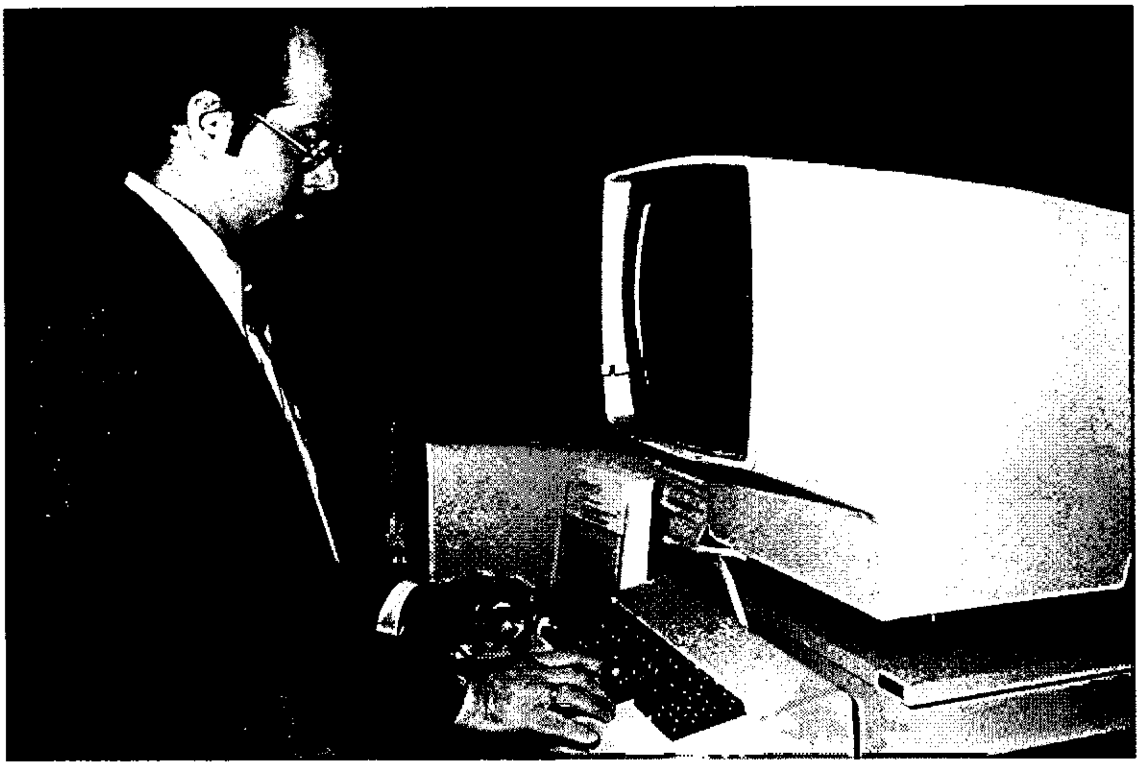
Het nieuwe systeem wordt onthuld

De afdeling Uitlening

De afdeling Informatie



De afdeling PTT-aanvragen



Terminal voor hal-aanvragen

Magazijnprinter

Publication 12

CICS/VS

Ervaringen met het Command-level van CICS/VS bij de ontwikkeling

van een geautomatiseerde bibliotheektoepassing

Experiences with COMMAND-level CICS/VS at the development of a computerised library application

1980

The designer designs and not the method.

Jaap van Rees [1982]

This article previously appeared in:

Informatie, April 1980, pages 313- 322

Abstract

In 1968, IBM announced the Customer Information Control System (CICS). This message control system is designed for dealing with transactions in a database / data communication environment. CICS is an interface between the operating system and a number of message processing programs. In the first version of CICS, programmers did not just have to know a computer language but also needed to be well-informed of a number of internal CICS affairs because manipulating with pointers, control blocks, and so on was an integral part of the application software.
In 1977/1978, in addition to the existing MACRO level version of CICS a COMMAND level version of CICS was announced, whicb meant that manipulation using control-blocks, pointers and so on was a thing of the past. At the start of 1978, the DUT Computer Centre of the Delft University of Technology procured CICS/VS with the COMMAND level interface to use this for computerisation of the Library. This article discusses a number of problems that appeared during the design and programming as well as the selected solutions in COMMAND level CICS/VS.

ERVARINGEN MET HET COMMAND-LEVEL VAN CICS/VS BIJ DE ONTWIKKELING VAN EEN GEAUTOMATISEERDE BIBLIOTHEEK-TOEPASSING

door ir. A. J. van Dijk

Keywords: library, AUBID, interface, CICS/VS, operating system
Kodes: 000.076.650
000.014.900
I

In 1968 annonceerde IBM het Customer Information Control System (CICS). Dit berichtenbesturingssysteem is ontworpen voor het behandelen van transakties in een data base/data kommunikatie omgeving. CICS is een interface tussen het operating system en een aantal berichtenverwerkingsprogramma's. Bij de eerste versie van CICS moesten de programmeurs niet alleen een programmeertaal kennen maar ook goed op de hoogte zijn van een aantal interne zaken van CICS omdat het manipuleren met pointers, control-blocks, enzovoort een belangrijk onderdeel was van de toepassingsprogramma's.
In 1977/1978 werd naast de bestaande MACRO-level versie van CICS een COMMAND-level versie van CICS geannonceerd waardoor het manipuleren met control-blocks, pointers enzovoort tot het verleden kon behoren.
In begin 1978 schafte het Rekencentrum van de Technische Hogeschool Delft ten behoeve van de automatisering van de Bibliotheek CICS/VS met het COMMAND-level interface aan.
De eerste fase van die automatisering is inmiddels voltooid. In dit artikel wordt ingegaan op een aantal problemen die bij het ontwerp en de programmering aan de orde zijn gekomen en de gekozen oplossingen in COMMAND-level CICS/VS.

1 INLEIDING

Het Customer Information Control System (CICS) is een berichtenbesturingssysteem, ontworpen voor het behandelen van transakties (logische eenheden te verwerken gegevens) in een data base/data kommunikatie omgeving.
Het is een interface tussen het operating system en een aantal berichtenverwerkingsprogramma's (toepassingsprogramma's). Hoewel CICS voor het operating system een toepassingsprogramma is, kan het worden beschouwd als een soort besturingssysteem waaraan de berichtenverwerkingsprogramma's worden toegevoegd (zie figuur 1). CICS is in 1968 door IBM geannonceerd. De eerste versie van CICS was alleen beschikbaar onder OS.
In 1971 verscheen een CICS-versie van DOS.
Van 1968 tot 1972 was het noodzakelijk om alle toepassingsprogramma's te schrijven in assembler language.
Naast kennis van assembler language was het noodzakelijk om zeer goed op de hoogte te zijn van een aantal interne zaken van CICS. Het manipuleren met pointers, control-blocks etc. was een belangrijk onderdeel van de toepassingsprogramma's.
Het programmeren geschiedde daarom in assembler language, aangevuld met CICS makro-instrukties.
In 1972 annonceerde IBM de mogelijkheid om toepassingsprogramma's voor CICS in ANSI-COBOL of PL/I te schrijven.
Het MACRO-level voor de CICS-opdrachten bleef echter gehandhaafd. In 1974 werd CICS vervangen door CICS/VS (Virtual Storage).
Eerst in 1977-1978 kwam naast het MACRO-level het COMMAND-level interface van CICS/VS beschikbaar. Programmeurs die met het COMMAND-level van CICS/VS werken hebben niet veel kennis nodig van de interne zaken van CICS/VS. Het manipuleren met control-blocks, pointers etc. behoort tot het verleden.
In begin 1978 schafte het Rekencentrum van de Technische Hogeschool Delft ten behoeve van de Bibliotheek van de Technische Hogeschool het programmatuursysteem CICS/VS aan.
In dit artikel wordt eerst in het kort iets gezegd over de toepassing. Daarna komen een aantal problemen van de toepassing en de gekozen oplossingen in COMMAND-level CICS/VS aan de orde.

2 DE TOEPASSING

In 1977 kwam het de Bibliotheek van de TH-Delft ter ore dat er in Dortmund een geïntegreerd bibliotheeksysteem was ontwikkeld, dat wellicht voor Delft interessant was. Het betrof hier het DOrtmund BIbliotheks-System (DOBIS). Dit systeem bleek inmiddels ook in gebruik te zijn bij de Katholieke Universiteit te Leuven (K.U.L.). De K.U.L. was bezig een aantal veranderingen en uitbreidingen aan te brengen. De K.U.L. noemde deze Leuvense versie van het Dortmundse systeem: LIBIS.
Nadat een werkgroep van de Bibliotheek de bi-

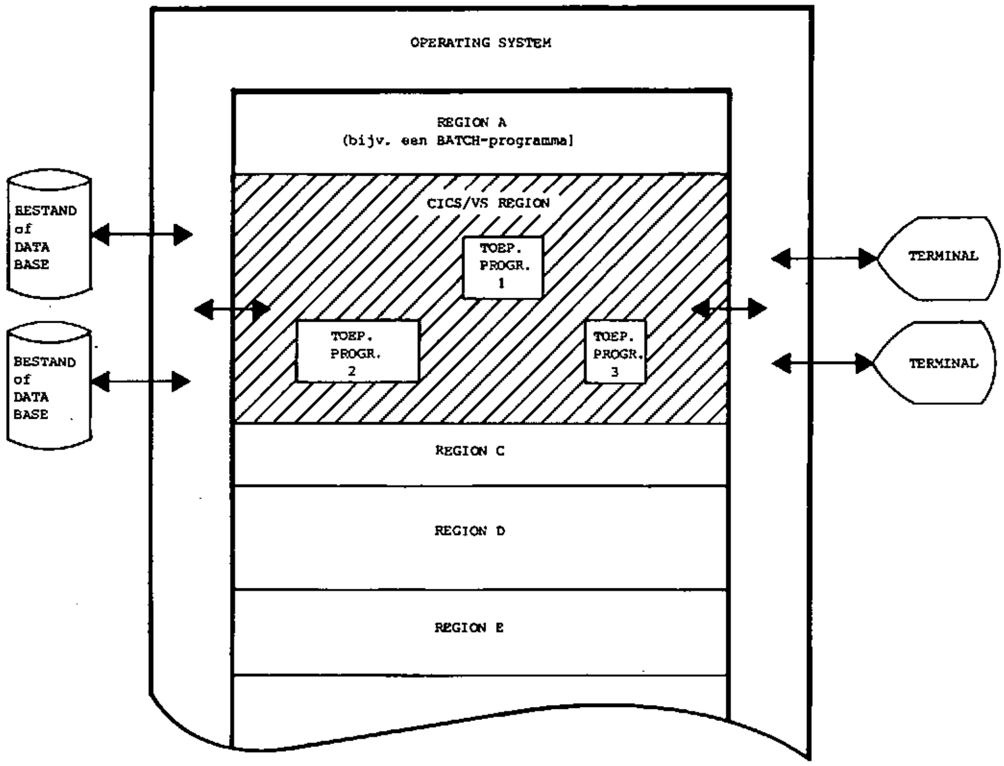


fig. 1: CICS/VS als interface tussen het Operating System en een aantal toepassingsprogramma's

bliotheektechnische aspekten van DOBIS had bekeken, werd het Rekencentrum van de TH-Delft gevraagd te onderzoeken of het mogelijk was het DOBIS en/of LIBIS systeem op het computersysteem van het Rekencentrum te implementeren. Na een preliminaire studie kon deze vraag bevestigend worden beantwoord.

De Bibliotheek van de TH-Delft bleek voorkeur te hebben voor het systeem LIBIS boven DOBIS. Op het moment dat de beslissing over aanschaf moest worden genomen was het systeem LIBIS echter nog niet geheel gereed. Eén van de kenmerken van de Delftse bibliotheek, die van het gesloten type is waardoor bezoekers niet in de boekenmagazijnen mogen komen, is dat de bezoekers zelf met behulp van de beschikbare apparatuur de gewenste boeken aanvragen.

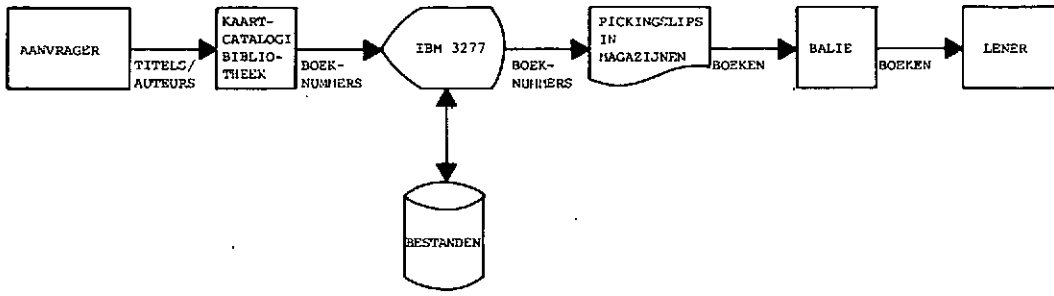
Programmatuur dat een en ander op een zeer gebruiksvriendelijke manier mogelijk maakt, is niet aanwezig in DOBIS en LIBIS.

Mede om deze redenen heeft de TH-Delft toen besloten om als tussenstap naar LIBIS zelf een intermediair aanvraagsysteem te ontwikkelen (zie figuur 2) en startte in 1977 het AUBID-projekt (AUtomatisch Bibliotheek Informatiesysteem Delft).

De voordelen van het intermediair aanvraagsysteem zijn:

a De aanvrager krijgt binnen enkele sekonden te zien of de gevraagde boeken al dan niet aanwezig zijn.
b Boeknummers van boeken die niet aanwezig zijn worden niet doorgegeven aan het boekenmagazijn. Voorheen werden alle boeknummers van aangevraagde boeken doorgestuurd naar het boekenmagazijn. Aangezien ± 40% van de aangevraagde boeken uitgeleend is, betekent het 'afvangen' van de aanvragen van boeken die uitgeleend zijn een forse werkbesparing voor het magazijnpersoneel, dat de vrijkomende tijd aan

fig. 2: Het afhandelen van een boekenaanvraag met het intermediair aanvraagsysteem



ander belangrijk werk in het boekenmagazijn kan besteden.

c De opbouw van de LIBIS-bestanden kan geleidelijk aan plaatsvinden. Het ligt in de bedoeling om de LIBIS-bestanden op te bouwen door de boeken die zijn uitgeleend, bij terugbezorging daarin op te nemen. Op die manier worden deze bestanden gevuld met het meest 'aktieve' deel van het boekenbezit.
Wanneer nu de LIBIS-bestanden voldoende omvang hebben en de betreffende funkties van het (aangepaste) systeem LIBIS geïmplementeerd zijn, kan worden overgeschakeld van het intermediaire aanvraagsysteem naar de LIBIS uitleenfunktie.

d Klanten en personeel van de Bibliotheek kunnen alvast wennen aan de IBM 3270 beeldschermterminals.

e De programmatuur van LIBIS is gebaseerd op het programmatuursysteem CICS/VS van IBM. Door nu de programmatuur van het intermediair aanvraagsysteem eveneens te baseren op CICS/VS heeft de projektgroep ervaring kunnen opdoen in het werken met CICS/VS.

f Het bibliotheekpersoneel is op korte termijn met daadwerkelijke resultaten gekonfronteerd en de verdere invoering van LIBIS kan stap voor stap plaatsvinden. Deze stap voor stap methode is zeer belangrijk, omdat het bij LIBIS gaat om een kompleet informatiesysteem. Door stap voor stap, funktie voor funktie te implementeren, kan de projektgroep de betreffende afdeling van de Bibliotheek de noodzakelijke begeleiding geven.

g Statistische gegevens komen snel beschikbaar.

h Er behoefde niet te worden gewacht op realisatie van de nog in bewerking zijnde funkties van LIBIS.

Voordat met de bouw van het intermediair aanvraagsysteem werd begonnen, zijn een aantal wensen geformuleerd waaraan het systeem moest voldoen. De volgende funkties moesten in het systeem aanwezig zijn:

- De boekstatus van ieder boek vastleggen.
  Zo'n boekstatus kan zijn: aanwezig, uitgeleend, niet beschikbaar, enzovoort.
- Van boeken met duplikaten het aantal exemplaren vastleggen.
- Boekaanvragen verwerken en aan de hand van de aanvragen de status van de betreffende boeken wijzigen.
- Van de aangevraagde boeken (ook voor duplikaten) die aanwezig blijken te zijn, het boeknummer en andere relevante informatie afdrukken op een matrixprinter in het betreffende boekenmagazijn.
- Terugbezorgingen verwerken, hetgeen onder andere betekent dat de status van de betreffende boeken moet worden gewijzigd.
- Van nieuwe boeknummers de boekstatus opnemen.
- Statistische gegevens met betrekking tot de gepleegde aanvragen verzamelen en overzichten produceren.

De bovenstaande wensen werden aangevuld met wensen die naar voren kwamen uit interviews met bibliotheekmedewerkers. De apparatuur in de Bibliotheek bestaat uit IBM 3277 beeldschermterminals en matrixprinters van het type IBM 3287. In de aanvraaghal, waar bezoekers de katalogi kunnen raadplegen en boeken kunnen aanvragen, staan vier beeldschermterminals. Daarnaast heeft de afdeling 'Uitlening' twee beeldschermterminals, één voor de uitleenbalie, waarnaar de aangevraagde boeken via een wentelgoot worden getransporteerd, en één voor de terugbezorgbalie, waar terugontvangen boeken worden verwerkt.
De afdeling 'Informatie', die midden in de aanvraaghal is gesitueerd, heeft één beeldschermterminal evenals de afdeling 'PTT', die de aanvragen die via de Post, de Telefoon en de Telex binnenkomen afhandelt.
De vier boekenmagazijnen beschikken elk over een matrixprinter waarop de gegevens van aangevraagde boeken die in het magazijn aanwezig zijn worden afgedrukt.
Eén van de boekenmagazijnen heeft tevens een beeldschermterminal waarmee de boekstatussen van nieuwe boeken kunnen worden vastgelegd.

3 CICS/VS IN RELATIE TOT DE APPLIKATIE

3.1 Algemeen

De programma's van het intermediair aanvraagsysteem zijn geschreven in COBOL en COMMAND-level CICS/VS release 1.4. In de navolgende punten worden enkele problemen, die tijdens het tot stand komen van de applikatie aan de orde zijn gekomen, en de daarbij behorende oplossingen besproken. Daarbij wordt er vanuit gegaan dat enkele basisbegrippen van CICS/VS bij de lezer bekend zijn.

3.2 Het opstarten van de applikatie

Door het intoetsen van de transaktiekode 'BIBL' kan het personeel van de Bibliotheek de beeldschermterminal bekend maken aan het intermediair aanvraagsysteem.
's Morgens zal het bibliotheekpersoneel alle beeldschermterminals via de transaktiekode 'BIBL' onder de applikatie brengen.
Hierbij kan zich een probleem voordoen. De eerste keer dat na het opstarten van CICS de transaktiekode 'BIBL' wordt ingetoetst worden er een aantal initialiseringshandelingen verricht waarbij o.a. enkele records van bestanden worden gelezen. Zo wordt bijvoorbeeld uitgezocht of het een normale (koude) start betreft of een EMERGENCY RESTART. De initialiseringshandelingen kosten enige tijd waardoor het mogelijk is dat een tweede, met behulp van 'BIBL' aan de applikatie bekend gemaakte, beeldschermterminal de transaktie al heeft afgehandeld omdat minder initialiseringshandelingen behoeven te worden verricht en beschikbaar is voor verdere applikatiegerichte handelingen. Dit leidt tot problemen. Om een en ander te voorkomen worden de transakties 'BIBL' geserialiseerd door middel van een semafoor.
De semafoor is in 'BIBL' gerealiseerd door in het begin van het bij 'BIBL' behorende programma een vast record van een bestand te lezen voor UPDATE en het

record pas na het voltooien van het initialiseringsproces vrij te geven.

3.3 Funktiescheiding

Het intermediair aanvraagsysteem is als het ware in vier stukken verdeeld. Deze vier stukken zijn de basisfunkties van het intermediair aanvraagsysteem. Iedere beeldschermterminal kan slechts één van deze vier funkties 'bedienen'. Het bibliotheekpersoneel heeft de gelegenheid om aan iedere beeldschermterminal de gewenste funktie toe te kennen. De vier mogelijke funkties zijn:

funktie 1: TERMINAL VOOR HALAANVRAGEN (NEDERLANDS)
funktie 2: TERMINAL VOOR HALAANVRAGEN (ENGELS)
funktie 3: TERMINAL VOOR INFORMATIEBALIE OF UITLEENBALIE
funktie 4: TERMINAL VOOR PTT-AANVRAGEN

Het is niet zonder meer mogelijk om van de ene funktie in de andere funktie te komen. Wel heeft het bibliotheekpersoneel de mogelijkbeid om via een wachtwoord een funktie te verlaten en opnieuw een funktie aan de onderhavige beeldschermterminal toe te kennen. Deze werkwijze heeft twee voordelen:

1 er is een duidelijke funktiescheiding bereikt; de aanvragers in de hal kunnen niet de bibliotheekfunkties bereiken
2 het systeem is flexibel met betrekking tot het gebruik; normaliter zullen 3 beeldschermterminals in de hal van funktietype 2 en één van funktietype 2 zijn. Wanneer nu een groep buitenlanders de Bibliotheek bezoekt (bijvoorbeeld kursisten van de internationale kursussen die aan de TH worden gegeven) kan het bibliotheekpersoneel één of meer beeldschermterminals in de aanvraaghal veranderen in een funktietype 2 terminal waardoor de dialoog in de engelse taal verloopt. Ook kan het bibliotheekpersoneel tijdelijk gebruik maken van een beeldschermterminal in de aanvraaghal door deze te veranderen in een funktietype 3 terminal. Dit laatste kan nodig zijn als plotseling één van de beeldschermterminals van het bibliotheekpersoneel uitvalt of als er erg veel werk aan de terminals gedaan moet worden en het in de aanvraaghal erg rustig is.

De funktiescheiding is mede gerealiseerd door alle transakties te beëindigen via een TRANSID-parameter in het RETURN-statement, waardoor is vastgelegd welke transaktie vanaf de betreffende beeldschermterminal wordt gestart nadat de gebruiker een volgende aktie heeft ondernomen (zie figuur 3). De gebruiker wordt op die manier volledig aan de hand genomen en waarnodig gekorrigeerd door mededelingen op het scherm.
Bovendien behoeft de gebruiker geen enkele transaktiekode te onthouden en in te toetsen. Alleen door een speciale serie handelingen te verrichten (o.a. een wachtwoord intoetsen) kan de applikatie worden verlaten en kan men transaktiekodes intoetsen. Deze serie handelingen is alleen bij bepaalde funktionarissen van de Bibliotheek bekend.

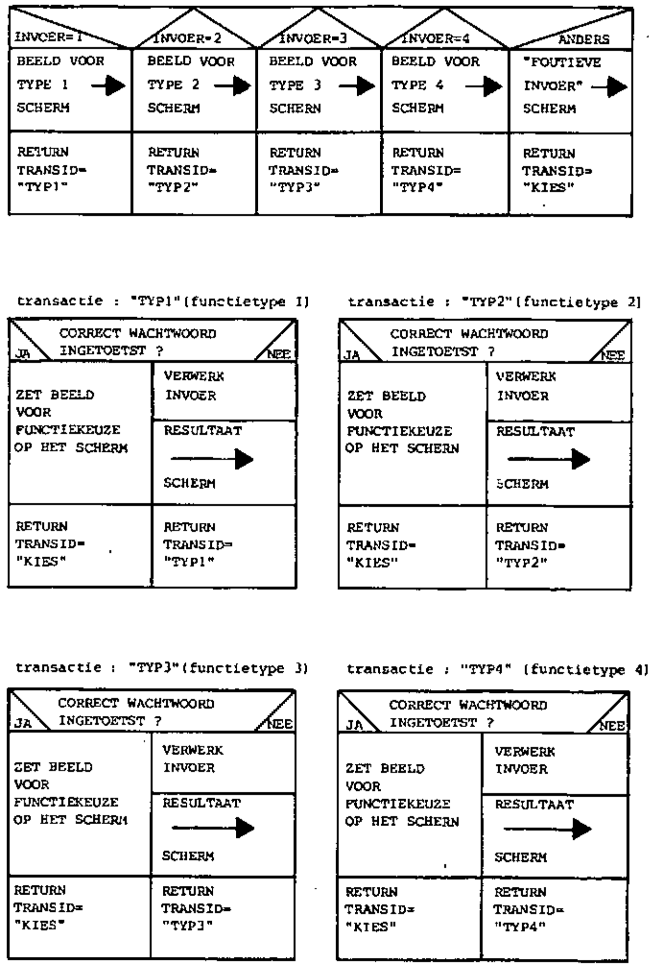


fig. 3: Funktiescheiding met behulp van een TRANSID-paramater in de RETURN-statements.

3.4 De printqueues voor de magazijnprinters

Nadat de bezoeker van de Bibliotheek, de katalogi geraadpleegd heeft en een formuliertje met de gewenste boeknummers heeft ingevuld, wordt hij vervolgens gekonfronteerd met de beeldschermterminal en de op het scherm van de terminal vermelde gegevens. De aanvrager moet nu de akties uitvoeren die op het scherm vermeld staan, te beginnen bij aktie 1, enz. Nadat de gewenste boeknummers zijn ingetoetst wordt op de enter-toets gedrukt en wordt de aanvraag in behandeling genomen. Voor ieder boek dat volgens het computerbestand 'AANWEZIG' is wordt een zogenaamde 'pickingslip' gemaakt. Daartoe wordt het boeknummer alsmede enkele andere gegevens zoals aanvraagnummer, aantal aanwezige boeken per aanvraag, magazijnnummer, pickingslipnummer en balienummer in een magazijnmededeling opgenomen. De mededelingen voor de magazijnen worden door de taak die de aanvraag afhandelt in transient-data-queues gezet. Ieder magazijn heeft een eigen queue. Deze queue wordt 'geleegd' door een taak die automatisch gestart wordt. Het automatisch starten gebeurt door middel van een trigger level. Als het aantal records in de queue gelijk wordt aan de waarde van het trigger

level wordt de taak gestart. Voor de printqueues geldt: trigger level = 1.
Een hoger trigger level is niet mogelijk omdat anders een aanvrager in sommige gevallen moet wachten tot er meerdere aanvragers zijn geweest alvorens hij zijn boek kan afhalen bij de uitleenbalie.
Een probleem doet zich voor wanneer de magazijnprinter niet goed funktioneert. De aanvrager wacht nu tevergeefs op zijn boeken. Het zou beter zijn als in die gevallen een mededeling op de funktietype 3 terminals van de afdelingen 'Informatie' en 'Uitlening' kwam dat de printer van magazijn ... niet werkt en moet worden nagezien. Het intermediair aanvraagsysteem bevat zo'n voorziening. Daartoe wordt in de automatisch gestarte taak via het START-commando een START met INTERVAL (2 minuten) 'uitgezet' voor iedere regel die op de printer moet worden afgedrukt (zie figuren 4a en 4b). Dit houdt in dat 2 minuten nadat de START is gegeven een taak zal worden opgestart die een mededeling verstuurt naar alle terminals van funktietype 3 (zg. 'WEKKER-taak').
Na het START-commando wordt de af te drukken regel verstuurd naar de matrixprinter. In dit SEND-commando wordt de parameter WAIT meegegeven. Dit impliceert dat het programma pas verder gaat als de SEND volledig is afgewerkt en de regel op de matrixprinter is afgedrukt.
Na voltooiing van de SEND-opdracht wordt nu een CANCEL-commando gegeven voor de start van de bovengenoemde WEKKER-taak, die daarmee weer van het toneel verdwijnt. Indien nu de magazijnprinter uit staat of om een andere reden niet goed funktioneert zal het programma dat de regel verstuurt in het SEND-commando 'blijven hangen'. Op die manier komt het CANCEL-commando niet aan bod en zal na 2 minuten de WEKKER-taak worden gestart die een boodschap verstuurt naar alle terminals van funktietype 3.
Dit laatste houdt in dat er voor iedere terminal van funktietype 3 een taak wordt gestart die op de onderste regel van het scherm de boodschap 'MAGAZIJN-PRINTER ... FUNKTIONEERT NIET, KONTROLEER ALLE PRINTERS' neerzet.
Daarvoor is het noodzakelijk dat van iedere beeldschermterminal het funktietype bekend is. In het Common Work Area wordt een en ander geadministreerd.

fig. 4a: De WEKKER-taak bij normaal verloop (de printer funktioneert)

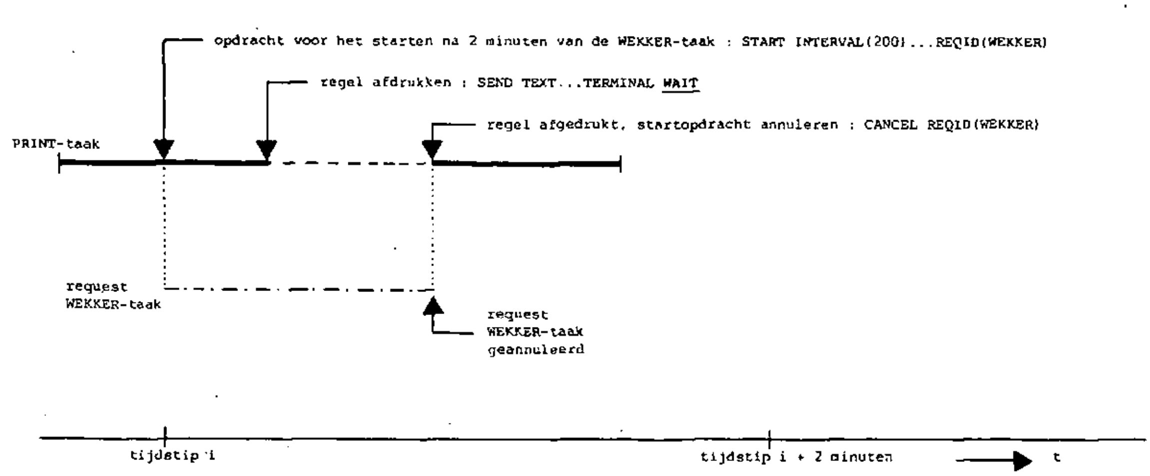
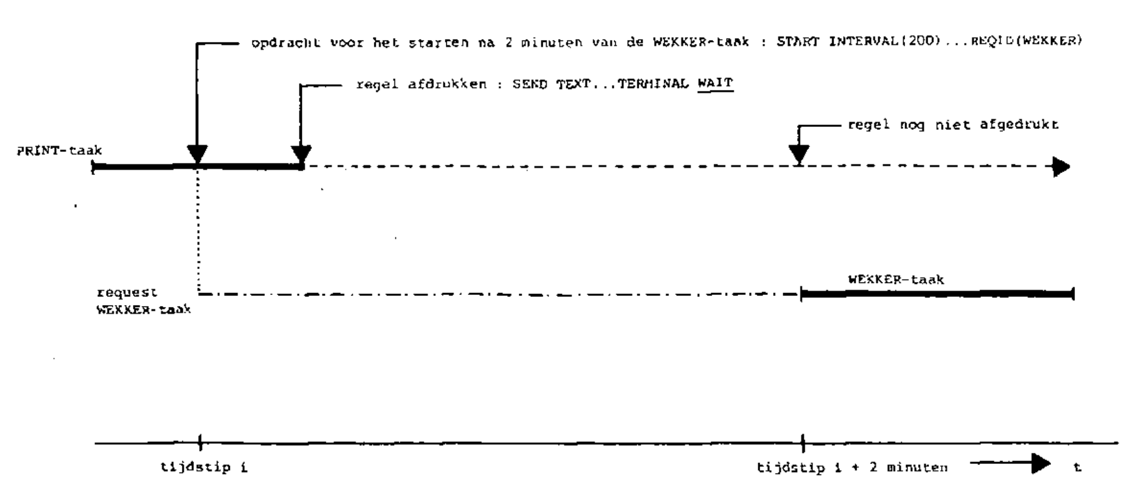


fig. 4b: De WEKKER-taak bij niet funktioneren van de printer



VOORBEELD : MAGAZIJN 2 SWITCHT VAN PRINTER 2 (DIE UIT STAAT) NAAR PRINTER 3

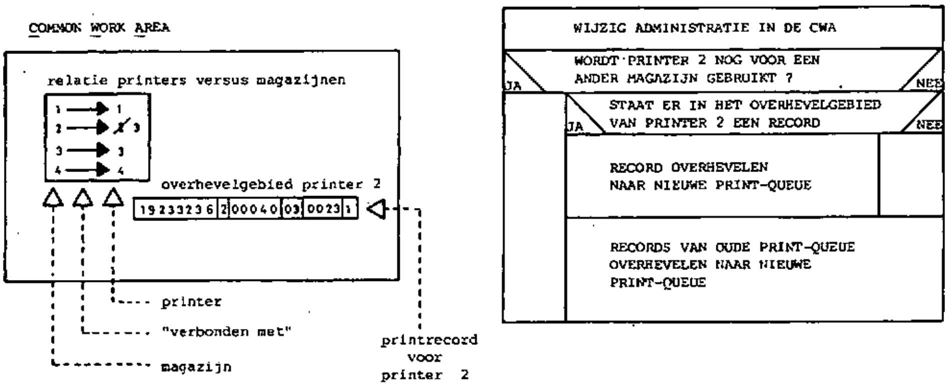


fig. 5: Het switchen van magazijnprinters

3.5 Het switchen van magazijnprinters

Normaliter is aan boekenmagazijn 1 matrixprinter 1 toegewezen, boekenmagazijn 2 ontvangt de aanvragen en boodschappen via matrixprinter 2, etc.
Soms is het gewenst een boekenmagazijn aan een andere matrixprinter te koppelen, bijvoorbeeld als een matrixprinter kapot is kunnen de aanvragen tijdelijk via een andere matrixprinter worden verstuurd. Ook uit bedrijfsorganisatorisch oogpunt is het soms wenselijk meerdere magazijnen te verbinden met één matrixprinter, bijvoorbeeld wanneer op stille tijden één magazijnmedewerker meerdere magazijnen bedient.
Met de subfunktie 'magazijninformatie' van funktietype 3 kan op ieder moment een willekeurige matrixprinter aan een magazijn worden gekoppeld. Dit switchen van matrixprinter is niet zo'n probleem als de oude matrixprinter in bedrijf is en blijft. De bij de matrixprinter behorende queue wordt dan gewoon geleegd (zie figuur 5).
Aanvragen voor het betreffende magazijn zullen vanaf het moment dat van matrixprinter verwisseld is in de queue van de 'nieuwe' matrixprinter terecht komen.
Wanneer de 'oude' matrixprinter niet meer werkt is er het probleem van de queue die nog een aantal aanvragen en/of boodschappen kan bevatten, omdat het aanvragen van boeken ook doorgaat als de printer niet werkt. Bovendien 'hangt' er aan die queue een taak die een aanvraag uit de queue naar de matrixprinter wil schrijven. Aangezien het mogelijk is om meerdere taken op dezelfde queue te laten werken wordt bij het verwisselen van matrixprinter gekeken of de oude matrixprinter nog gebruikt wordt door een ander magazijn. Als geen enkel magazijn de matrixprinter nog gebruikt, wordt een zogenaamde 'HEVEL-taak' gestart die de records uit de oude queue overhevelt naar de nieuwe queue.
Een probleem daarbij is dat het record dat reeds via de (trigger level-)taak onderweg was naar de oude matrixprinter niet wordt overgeheveld. Dit is niet acceptabel omdat een aanvrager in de aanvraaghal op die manier nooit zijn aangevraagde boek krijgt. Om dit op te lossen wordt ieder record dat via de (trigger level-)taak naar een matrixprinter wordt verstuurd eerst in een hulpgebied in de Common Work Area geplaatst. Nadat de aanvraag is afgedrukt wordt dit hulpgebied op blanks gezet.
De HEVEL-taak zal voordat gestart wordt met hevelen in het hulpgebied kijken of er nog een record staat dat als eerste naar de nieuwe queue moet worden getransporteerd.

3.6 Het versturen van terminalmededelingen

De subfunktie 'terminalmededeling versturen' van funktietype 3 kan gezien worden als een 'intercom' mogelijkheid. Vanaf een funktietype 3 terminal kan een boodschap worden verstuurd naar alle terminals van funktietype 1, naar alle terminals van funktietype 2, naar alle terminals van funktietype 3, naar alle terminals van funktietype 4 of naar alle beeldschermterminals.
De boodschap wordt op de onderste regel van de betreffende beeldschermterminals zichtbaar gemaakt met een hoge lichtintensiteit. Het personeel van de Bibliotheek heeft de mogelijkheid om een tiental mededelingen die regelmatig voorkomen in het systeem te laten opnemen. Zodra een mededeling verstuurd moet worden, verschijnen deze vastgelegde mededelingen op het scherm en is het intoetsen van een getal tussen 1 en 10 voldoende om de gewenste mededeling te versturen. Eenmalige mededelingen kunnen vanzelfsprekend ook worden verstuurd. De vastgelegde mededelingen kunnen op ieder gewenst moment gewijzigd worden.
Verstuurde mededelingen zullen onmiddellijk op de betreffende schermen verschijnen en na ieder aktie vanachter die beeldschermterminals opnieuw verschijnen. Alleen door het sturen van een blanko mededeling is de oude mededeling te verwijderen. Het kontinueren van de mededeling op een terminal is mogelijk, omdat de mededeling niet alleen 'aktief' wordt verstuurd en derhalve onmiddellijk op het scherm verschijnt maar ook 'passief' wordt verstuurd.
Dit passief versturen houdt in dat de mededeling wordt geplaatst in het mededelingengebied van funktietype 1, funktietype 2, funktietype 3 en/of funktietype 4. Deze mededelingengebieden staan in de Common Work Area. Wanneer nu een MAP (dit is

een tabel waarin alle velden van een bepaalde schermindeling met symbolische namen worden gedefinieerd; deze MAP wordt met behulp van Basic Mapping Support van CICS/VS omgezet in een 3270 data stream) naar een beeldschermterminal moet worden verstuurd, zal afhankelijk van het funktietype één van de mededelingen uit de Common Work Area mee worden verstuurd.
Op deze manier is het bijvoorbeeld mogelijk om een mededeling de gehele dag door op het scherm te laten verschijnen.
Een probleem bij het aktief versturen van de mededelingen is dat wanneer de mededeling via een afzonderlijke MAP wordt verstuurd, het niet mogelijk is gebleken de 'CURSOR' ongemoeid te laten. Omdat de cursor meestal in een 'protected field' terecht komt krijgt de beeldschermterminalgebruiker te maken met een geblokkeerd toetsenbord.
Vooral voor een onervaren gebruiker is dit erg vervelend gebleken.
Geprobeerd is om bij het verzenden de parameter CURSOR zonder nadere specifikatie mee te geven in de hoop dat daardoor de cursor ongemoeid zou worden gelaten. Helaas bleek dat de cursor in dat geval naar positie 1 van het scherm wordt verplaatst.
Een oplossing die wel blijkt te werken is het zelf opbouwen van een datastream voor de IBM 3270 in de DATA DIVISION van het COBOL-programma. Dit impliceert echter wel dat de programmeur, die op COMMAND-level CICS/VS werkt zich moet verdiepen in de datastream van de IBM 3270 en dus moet gaan werken met instrukties als SF (Start Field), SBA (Set Buffer Address), etc. Deze instrukties zijn via bit-manipulatie in de DATA DIVISION van een COBOL-programma op te nemen.
Het ontbreken van een mogelijkheid om bij het gedeeltelijk beschrijven van een scherm op COMMAND-level CICS/VS de cursor ongemoeid te laten is daarom enigszins merkwaardig.

3.7 Het gebruik van de CICS/VS master-terminal-transaktie

Ten behoeve van het operationele gebruik beschikt CICS/VS over een 'master-terminal-transaktie' genaamd 'CSMT'.
Met behulp van deze transaktie kan bijvoorbeeld een lijn IN SERVICE worden gezet : CSMT LIN, INSRV, TERMID = PR40
Ook kunnen terminals IN SERVICE of OUT OF SERVICE worden gezet, taken worden afgebroken, trigger levels worden gewijzigd, enzovoort.
Er is bij het realiseren van het intermediair aanvraagsysteem naar gestreefd het systeem zoveel mogelijk 'self-supporting' te laten zijn. Enerzijds omdat het de eerste CICS/VS applikatie was en in de computerzaal geen master-operator-terminal voor CICS/VS beschikbaar was en anderzijds omdat het personeel van de service-diensten van de Bibliotheek geen automatiseringskennis bezat. Aangezien de Bibliotheek ook op enkele avonden en op zaterdagochtend geopend is en de computer van het Rekencentrum op zaterdag alleen 'onbemand' beschikbaar is moet het bibliotheekpersoneel het intermediair aanvraagsysteem kunnen afsluiten.
Om bovenstaande redenen was het zeer gewenst om CSMT-transakties vanuit een programma te kunnen laten uitvoeren. Het laten uitvoeren van een CSMT-transaktie is gelukt door deze transaktie, begeleid door een audible-alarm en de boodschap 'DRUK OP DE ENTER-TOETS', op het scherm te zetten. Zodra een personeelslid op de enter-toets drukt wordt de transaktie uitgevoerd.
Het laten uitvoeren van een CSMT-transaktie zonder tussenkomst van een terminal-operator is niet gelukt onder COMMAND-level CICS/VS hetgeen erg jammer is. Wel is het gelukt door met een subroutine op MACRO-level te werken.

3.8 Het ketenen van CSMT-transakties

Zoals vermeld, sluit het personeel van de Bibliotheek zelf de applikatie af. Na het intoetsen van een geldig wachtwoord komt het afsluitingsproces op gang. De laatste handeling die moet worden verricht is het afsluiten van CICS/VS door middel van de transaktie CSMT SHU, NO
Alvorens deze transaktie kan worden uitgevoerd moeten een aantal handelingen worden verricht zoals het beëindigen van lopende taken. Een en ander houdt in dat er een aantal CSMT-transakties in de juiste volgorde moet worden uitgevoerd. Op COMMAND-level kunnen ze alleen worden uitgevoerd door tussenkomst van een terminal-operator.
Het intermediair aanvraagsysteem bevat enkele konstrukties van geketende CSMT-transakties.
Met name in het afsluitingsproces wordt het personeel van de Bibliotheek een aantal malen uitgenodigd om op de enter-toets te drukken. De konstruktie is zo gemaakt dat iedere 5 sekonden wordt gekontroleerd of de onderhavige CSMT-transaktie is uitgevoerd. Als de CSMT-transaktie is uitgevoerd wordt de volgende CSMT-transaktie, begeleid door een audible-alarm, op het scherm gezet. Wanneer de CSMT-transaktie nog niet is uitgevoerd, wordt dezelfde transaktie, eveneens begeleid door een audible-alarm, op het scherm gezet (zie figuur 6). Op deze wijze is het mogelijk om bibliotheekpersoneel dat geen enkele CSMT-transaktie kent lijnen en terminals OUT OF SERVICE te laten zetten, taken te laten beëindigen en zelfs CICS/VS te laten afsluiten. Sinds 31 oktober 1979, de dag waarop het systeem door de Bibliotheek in gebruik is genomen heeft zich op dit punt geen enkel probleem voor gedaan.
Het personeel van de Bibliotheek beschikt verder over 5 transaktiekodes die allemaal een keten van CSMT-transakties aktiveren. Zo zorgt bijvoorbeeld de transaktiekode OUTH er voor dat alle terminals die in de aanvraaghal staan OUT OF SERVICE worden gezet. Dit is van belang als sommige afdelingen van de Bibliotheek, zoals de afdeling 'Informatie' en de leeszalen, geopend zijn maar er geen boeken worden uitgeleend.
De konstruktie van de ketens is dusdanig dat beïnvloeding van de inhoud van de CSMT-transakties door een terminal-operator uitgesloten is. Alleen het tijdstip van uitvoeren kan worden beïnvloed. De tussenkomst van een terminal-operator kan worden voorkomen door gebruik te maken van een subroutine op MACRO-level.
Deze subroutine is het enige programma van de applikatie waarin het MACRO-level is gebruikt omdat

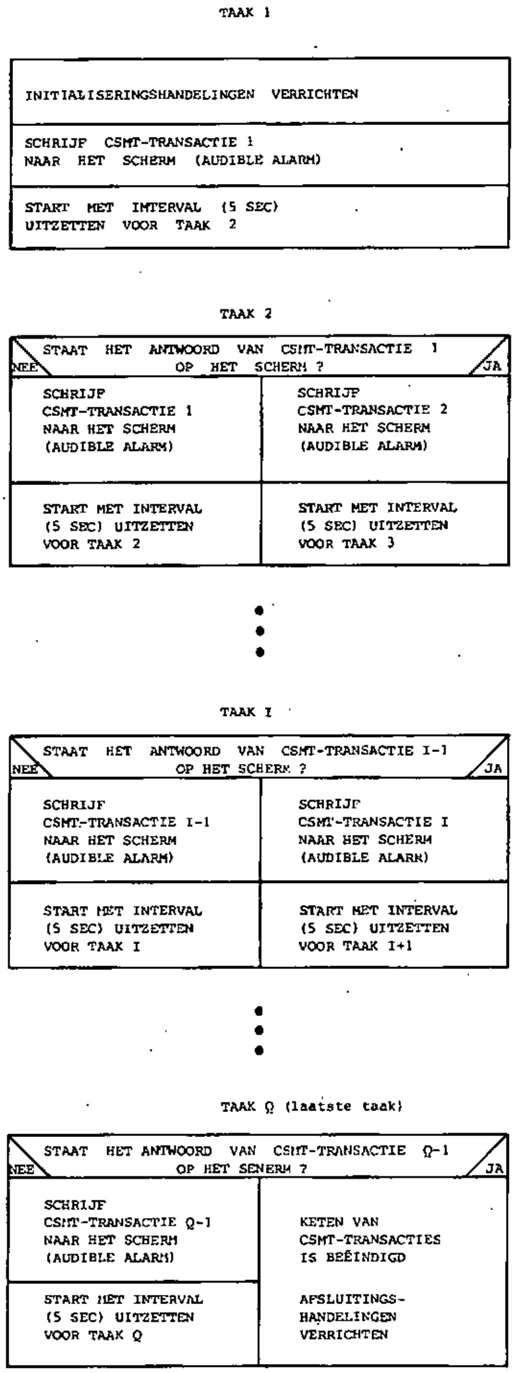


fig. 6: Het ketenen van CSMT-transacties

het probleem op COMMAND-level niet oplosbaar leek. De subroutine wordt o.a. gebruikt in een transaktie die 's morgens na het opstarten van CICS/VS kan worden gestart en die gedurende de gehele CICS/VS run om de 5 minuten alle bij de applikatie behorende lijnen en terminals kontroleert en eventueel IN SERVICE zet. Daardoor worden lijnen en/of terminals die door een of andere oorzaak OUT OF SERVICE zijn geraakt, bijvoorbeeld door een I-O-error, binnen enkele minuten automatisch IN SERVICE gezet.

De kwetsbaarheid van het intermedair aanvraagsysteem dat een belangrijk onderdeel is van het bedrijfsproces binnen de servicediensten van de Bibliotheek is daardoor belangrijk verminderd.

3.9 Recovery/restart

3.9.1 *De Common Work Area (CWA)*

De Common Work Area (CWA) is een werkgebied dat door alle toepassingsprogramma's gebruikt kan worden om gegevens op te slaan of op te halen.
In het intermediair aanvraagsysteem wordt de CWA gebruikt om te administreren welke terminals via 'BIBL' onder de applikatie bekend zijn gemaakt en van welk funktietype ze zijn. Daarnaast bevat het de mededelingen voor de vier funktietypen, de relaties magazijnen versus magazijnprinters, tellers voor de pickingslips, magazijnboodschappen enzovoort.
Een probleem is dat de CWA niet recoverable is.
In het intermediair aanvraagsysteem is het wenselijk dat het CWA-gebied recoverable is zodat na een 'system-down' of een 'CICS-down' met de oude gegevens verder gegaan kan worden.
Om dit te realiseren is een direkt-toegankelijk bestand gekreëerd. Aan het einde van iedere taak die de CWA modificeert wordt er een record, dat naast de laatste gegevens van de CWA datum en tijdstip bevat, naar het CWA-bestand geschreven. Tijdens het startproces worden aan de hand van datum en tijdstip de meest recente gegevens van de CWA opgezocht en in de CWA geplaatst.
Het aantal records van het CWA-bestand mag niet te klein zijn omdat anders wachttijden kunnen optreden. Deze wachttijden treden op wanneer bij een te klein aantal records, bijvoorbeeld 1 of 2 records, meerdere taken de gegevens in hetzelfde record van het CWA-bestand willen bijwerken.
Het CWA-bestand van het intermediair aanvraagsysteem bevat 25 records die beurtelings ('wrap around') worden bijgewerkt. Record 0 heeft een speciale betekenis (zie 3.9.2).

3.9.2 *Emergency Restart*

Het starten van CICS/VS door de operateur achter het systeem-console kan op verschillende manieren gebeuren.
Naast een 'koude start' bestaat er een 'warme start', een 'gedeeltelijk warme start' en een 'EMERGENCY RESTART'.
Aangezien de operateurs in wisseldiensten werken en ook in de Bibliotheek van wisseldiensten sprake is kan worden verwacht dat er ten gevolge van kommunikatiestoornissen af en toe een verkeerde startmethode wordt gebruikt.
Als er in plaats van een EMERGENCY RESTART een koude start plaatsvindt kunnen komplete transient-data-output-queues verdwenen zijn, hetgeen uitermate vervelend is. Om deze problematiek te omzeilen kent de operateur slechts één startcommando. Dit startcommando impliceert een EMERGENCY RESTART. Het intermediair aanvraagsysteem bepaalt zelf of het een 'echte' EMERGENCY RESTART is of dat er sprake is van een koude start. In het geval van een koude start wordt o.a. een DELETE-opdracht gegeven voor de transient-data-output-queues en

wordt de CWA op een andere manier geïnitialiseerd dan bij een EMERGENCY RESTART.
Het bepalen of het al dan niet een 'echte' EMERGENCY RESTART betreft gebeurt aan de hand van gegevens die in record 0 van het CWA-bestand (zie 3.9.1) staan. Die gegevens zijn er ingezet door een programma dat tijdens de laatste fase van het afsluitingsproces van CICS/VS wordt uitgevoerd.

3.9.3 *Transient-Data-Files*
Transient-data is een algemene queueing-faciliteit van CICS/VS. De intrapartition-transient-data queues (ITD-queues) zijn in CICS/VS vooral beschikbaar ten behoeve van sekwentiële uitvoerbestanden, die extrapartition-transient-data queues (ETD-queues) worden genoemd. Het intermediair aanvraagsysteem heeft de beschikking over ITD-queues ten behoeve van de magazijnprinters, het statistiekbestand en het mutatiebestand.
Iedere magazijnprinter heeft zijn eigen ITD-queue die via een trigger level een taak kan aktiveren die de records uit de ITD-queue naar de magazijnprinter verstuurt.
De ITD-queues ten behoeve van het statistiekbestand en het mutatiebestand kunnen eveneens van een trigger level worden voorzien en door middel van automatisch geïnitieerde taken (ATI) de records naar ETD-queues wegschrijven.
In het intermediair aanvraagsysteem is dat echter niet gebeurd. Beide ITD-queues zijn van de specifikatie 'recoverable' voorzien omdat met name aan het mutatiebestand, dat een interface-funktie naar een batchprogrammatuursysteem vervult, hoge betrouwbaarheidseisen zijn gesteld.
Zouden nu door middel van een ATI-taak de records naar een ETD-queue worden weggeschreven dan kunnen er problemen ontstaan na een CICS/VS-down of een system-down.
De ETD-queues kunnen namelijk niet van de specifikatie 'recoverable' worden voorzien, zodat na een CICS/VS-down of een system-down de ETD-queues records kunnen bevatten die ze in verband met het recovery mechanisme op de ITD-queues niet zouden mogen bevatten.
Er zijn 2 mogelijkheden om een en ander te voorkomen:

a in plaats van ETD-queues gebruik maken van andere bestanden die onder CICS/VS wel recoverable zijn. In dat geval kan via het ATI-mechanisme van tijd tot tijd een hoeveelheid gegevens worden overgeheveld naar een direkt-toegankelijk recoverable bestand. Deze methode verdient de voorkeur als de ITD-queues erg veel records bevatten;
b de gegevens van de ITD-queues worden gedurende de gehele dag opgespaard en tijdens het afsluiten van de applikatie overgeheveld naar ETD-queues. Een probleem hierbij is dat nadat de overheveltaak korrekt beëindigd is, er nog een CICS/VS-down of een system-down kan optreden voordat CICS/VS de ETD-queue heeft afgesloten. In die situatie is namelijk de ITD-queue niet meer beschikbaar omdat de overheveltaak korrekt is afgesloten en de ETD-queue is niet leesbaar voor batch-programma's.
Ten behoeve van deze situaties is het gewenst de ITD-queue te behouden.

In het intermediair aanvraagsysteem is mogelijkheid *b* toegepast. De ITD-queue wordt behouden door in de overheveltaak allereerst het einde van de ITD-queue aan te geven door middel van een sluitrecord en vervolgens ieder record dat naar de ETD-queue wordt geschreven ook nog eens te schrijven naar de ITD-queue waaruit het record gelezen is. Nadat de overheveltaak, die stopt na het lezen van het sluitrecord, is beëindigd is de ITD-queue nog beschikbaar. In de volgende CICS/VS-start wordt uitgezocht wat er met de ITD-queue moet gebeuren (zie 3.9.2).
Een nadeel van deze methode is dat het afsluitingsproces een aantal minuten in beslag kan nemen. Tijdens de operationele periode van het intermediair aanvraagsysteem is het tot nu eenmaal voorgekomen dat er een CIVS/VS-down optrad nadat de overheveltaak suksesvol beëindigd was en CICS/VS de ETD-queue nog niet had afgesloten.

3.9.4 *Dead-lock situaties*
De klassieke dead-lock situatie kan zich voordoen als taak 1 eerst record A en vervolgens record B van een bestand wil bijwerken en taak 2 eerst record B en vervolgens record A wil bijwerken. Als nu taak 1 record A via een exklusive control mechanisme vasthoudt en taak 2 record B vasthoudt is de dead-lock een feit. Dit vasthouden gebeurt in verband met het recoverable zijn van het bestand en het Dynamic Transaction Backout mechanisme dat voor de betrokken taken geldt. Het Dynamic Transaction Backout mechanisme zorgt er namelijk voor dat bij abnormale beëindiging van de taak de door deze taak aangebrachte veranderingen in het recoverable bestand te niet worden gedaan.
De dead-lock situatie wordt in het geval van het intermediair aanvraagsysteem voorkomen door de boeknummers die met de beeldschermterminal worden ingebracht te sorteren voordat ze verder worden verwerkt.
Een andere dead-lock situatie kan zich voordoen tussen bestanden. In het intermediair aanvraagsysteem zijn er transakties die onder Dynamic Transaction Backout werken en die 4 recoverable bestanden kunnen bijwerken. Het is van groot belang dat alle transakties de gewenste bestanden in dezelfde volgorde benaderen aangezien zich anders dead-lock situaties tussen bestanden kunnen voordoen.
Tijdens de bouw van het intermediair aanvraagsysteem was in de meeste transakties aan deze hiërarchieke bestandsbenadering voldaan. In enkele transakties echter was de volgorde waarin de 2 transient-data-output-queues werden benaderd niet overeenkomstig de volgorde in de andere transakties. Tijdens de eerste testfase waarin de bestanden nog niet de specifikatie 'recoverable' gekregen hadden heeft zich geen enkele dead-lock situatie voorgedaan. In de tweede fase waren de bestanden wel voorzien van de specifikatie 'recoverable'. Dit heeft geleid tot enkele dead-lock situaties. Nadat de transakties die niet konsekwent in hiërarchieke volgorde de bestanden bijwerkten waren gemodificieerd waren de deadlocks verdwenen.
De twee transient-data-output-queue in het inter-

mediair aanvraagsysteem betreffen een statistiekbestand en een mutatiebestand. Zodra een statistiekrecord is weggeschreven naar de betreffende transient-data-output-queue wordt het bestand geblokkeerd voor andere transakties totdat de onderhavige transaktie beëindigd is. Daarom is het verstandig om de gegevens voor de transient-data-output-queues voor de duur van de transaktie op te sparen en deze gegevens pas aan het einde van de transaktie, in de hiërarchieke volgorde, naar de transient-data-output-queues te schrijven.

4 SLOTOPMERKING

Het in dit artikel genoemde intermediair aanvraagsysteem is ontwikkeld door een projektgroep waarin zowel personeelsleden van de Bibliotheek als personeelsleden van het Rekencentrum participeerden. De oplossingen van de geschetste problemen met betrekking tot CICS/VS zijn gerealiseerd door ir. A. J. van Dijk van het Rekencentrum en J. C. Kwakernaak van de Bibliotheek. Aan de diskussies hebben ook A. P. P. van Buchem (IBM), drs. J. W. J. Heijnsdijk (Rekencentrum) en A. van der Meer (Rekencentrum) een bijdrage geleverd.

LITERATUUR:

David, I. W.
CICS/VS recovery/restart performance implications
TeleSystems Journal, Vol. V(1978) No. 3 pp. 12-14.

Davies, K. E.
CICS/VS High Level Programming Interface
IBM UK Laboratories Ltd., Hursley, England, Technical Bulletin G320-6033. April 1979.

IBM
Customer Information Control System/Virtual Storage (CICS/VS), Version 1, Release 4, General Information
IBM UK Laboratories Ltd., Hursley, England, GC33-0066-3. September 1978.

IBM
Customer Information Control System/Virtual Storage (CICS/VS), Version 1, Release 4, Application Programmer's Reference Manual (Command Level)
IBM UK Laboratories Ltd., Hursley, England, SC33-0077-1. Juni 1978.

IBM
Customer Information Control System/Virtual Storage (CICS/VS), Version 1, Release 4, System/Application Design Guilde
IBM UK Laboratories Ltd., Hursley, England, SC33-0068-1. Juni 1978.

Dijk, A. J. van
Delftse studenten vragen computer om boeken uit de bibliotheek DP-Monitor, 4e jaargang No. 1. Mei 1979 pp. 16-23.

Dijk, A. J. van, J. C. Kwakernaak, A. van der Meer
AUBID-projekt, fase 1, Systeemontwerp en programmabeschrijving Delft, januari 1980, 180 pag.

Publication 13

VERKEERSDATAVERZAMELING

Een automatisch systeem voor registratie van verkeersdata

en verwerking tot voertuigtrajectoriën

Traffic Data Collection
Equipment and a software system which derives vehicle trajectories
from the detector records on the magnetic tape

1975

The statement "Half of computerisation is achieved using a load of big fat zeroes!" is within the domain of information systems only applicable to the computer code and should not be applied to designers, programmers and users.

Aart van Dijk

This article previously appeared in:

Verkeerskunde, June 1975, pages 308- 314

Abstract
In behalf of the research of traffic streams it is desirable to collect data about traffic streams.
The Transportation Research Laboratory from the Department of Civil Engineering of the Delft University of Technology, has the equipment to carry out traffic measurements. With this equipment and with the help of traffic detectors the points of time of the activation of the detectors by the vehicle axes are recorded on magnetic tape.
This article describes the equipment and a software system which derives vehicle trajectories from the detector records on the magnetic tape.

A. J. van Dijk, Laboratorium voor Verkeerskunde, TH-Delft
Ir. D. Mozes, Dienst Gemeentewerken, 's-Gravenhage)*

Verkeersdataverzameling

Een automatisch systeem voor registratie van verkeersdata en verwerking tot voertuigtrajectoriën.

Ten behoeve van onderzoek naar de afwikkeling van verkeersstromen op wegen is het gewenst gegevens omtrent verkeersstromen te verzamelen. Het Laboratorium voor Verkeerskunde, dat een onderdeel is van de Afdeling der Civiele Techniek van de TH te Delft, beschikt over apparatuur om verkeersmetingen te verrichten. Met behulp van wegdetectoren worden door deze apparatuur de passagetijdstippen van voertuigassen op een magneetband vastgelegd. Dit artikel beschrijft zowel deze apparatuur als een programmatuursysteem dat uit de detectormeldingen voertuigtrajectoriën vervaardigt.

1. Inleiding

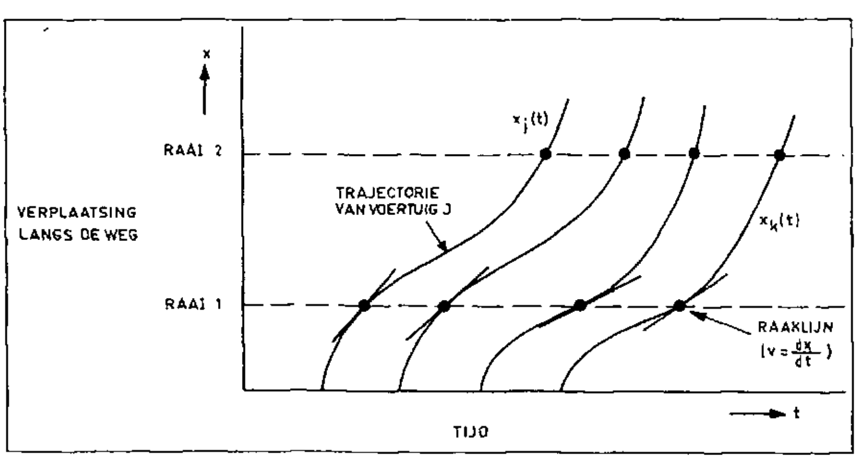
De Groep Verkeerstechniek van het Laboratorium Voor Verkeerskunde verricht speurwerk naar de afwikkeling van verkeersstromen op wegen, kruispunten e.d. Dergelijk onderzoek heeft tot doel om door middel van de vaststelling van het rijgedrag enerzijds en de invloed van de factoren mens, voertuig, weg, e.d. daarop anderzijds, te komen tot eén beter ontwerp van diverse wegelementen zoals kruispunten, invoeg- en uitrijpunten, enz. Verder tracht men hierdoor informatie te verkrijgen voor een beheersing van de verkeersstromen. Immers de snelle toeneming van de omvang van de investeringen in verkeersvoorzieningen vereist naast een doelmatige aanleg ook een zo goed mogelijk gebruik van deze voorzieningen.
Verkeersstroommodellen beschrijven het gedrag van de mens/voertuig combinatie in het verkeer. De complexiteit van het menselijk gedrag beperkt echter de toepasbaarheid van deze abstracte modellen bij het oplossen van practische problemen. Door het leveren van meer gedetailleerde data over verkeersstromen kunnen de modellen meer in overeenstemming gebracht worden met de werkelijkheid. De hiermee verband houdende verkeersstroommetingen zouden in principe alles moeten registreren wat van invloed kan zijn op het voertuiggedrag. Voorlopig is echter gekozen voor de beperking tot het vastleggen van de *voertuigtrajectoriën*. Met een voertuigtrajectorie wordt bedoeld (zie (1) en (2)) een tijd-weg-diagram waarin is vastgelegd de wegordinaat x als functie van de tijd t (zie afb. 1). De snelheid is bepaald door de formule $v = dx/dt$. Vaak bevat een tijd-weg-diagram de trajectoriën van alle voertuigen die zich gedurende een bepaald tijdvak op een wegvak bevinden. Met $x_j(t)$ wordt dan de trajectorie van voertuig j bedoeld.

In de afgelopen jaren zijn door het Laboratorium Voor Verkeerskunde met behulp van time-recorders metingen verricht. Zowel de waarneming met behulp van conventionele time-recorder-apparatuur, die de detectormeldingen op wasrollen registreert, als die middels filmregistratie vanuit een hoge positie, hebben als groot nadeel de tijdrovende en zeer arbeidsintensieve verwerking die grotendeels met de hand moet geschieden. Het uitwerken van de wasrollen van één meetdag vergt gemiddeld één manjaar. De onderzoeker moet dus (te) lang wachten voor hij de meetresultaten ter beschikking heeft.
Deze factoren leidden er toe dat in 1971 besloten werd de registratie en de verwerking van de benodigde meetgegevens verregaand te automatiseren. Hoewel met filmen veel meer informatie wordt vastgelegd is de verwerking hiervan niet volledig te automatiseren. Door gebruik te maken van wegdetectoren is dit ideaal eerder te bereiken.
Na een inventarisatie van de mogelijkheden werd gekozen voor een electronisch, automatisch en mobiel verkeersregistratiesysteem dat aan de vereiste (hoge) nauwkeurigheid voldeed en dat bestaat uit de volgende basiselementen:
— het detectiesysteem, bestaande uit detectoren en versterkers;
— een detectorsignaalmonitor (DSM); deze zet de detectorsignalen om in detectorinformatie die verwerkt kan worden door:
— de Raytheon 704 computer, die de ingevoerde informatie met behulp van programmatuur efficiënt wegschrijft op:
— een magneetbandeenheid, die IBM-compatibel is (dit in verband met de bij de TH-Delft aanwezige IBM 370/158 computer).

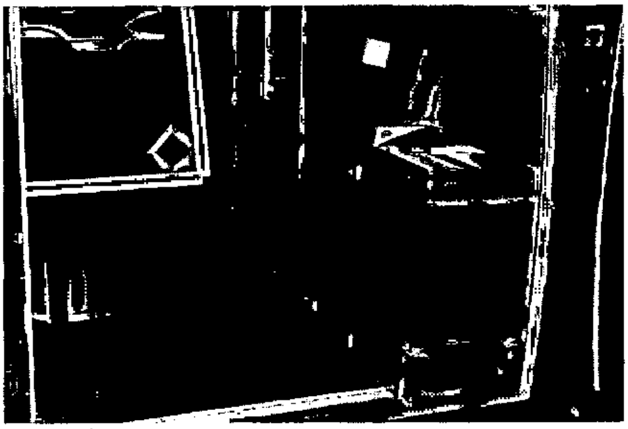
De wielen van passerende voertuigen reken op de weg gelegde detectoren (bijv. coaxiaalkabels) die dan signalen afgeven. Deze worden verstrekt en als elektrische puls ingevoerd in de meetwagen waarna deze signalen door de DSM omgezet worden in detectorinformatie die door de Raytheon 704 computer verwerkt wordt (zie afb. 2). Deze computer bepaalt van alle ingevoerde detectormeldingen het tijdstip van binnenkomst, waarna deze tijdstippen samen met de detectorinformatie op de magneetband worden vastgelegd.

De apparatuur, waarop 127 wegdetectoren kunnen worden aangesloten bepaalt die tijdstippen met een instelbare nauwkeurigheid van 125, 250, 500, 1000 of 2000 usec. Na deze verkeersdataregistratie vindt de verkeersdataverwerking plaats. Met computerprogrammatuur worden de oorspronkelijke „voertuigen" gevormd uit de detectormeldingen waarna de voertuigtrajectoriën samengesteld worden door dezelfde voer-

* Ir. D. Mozes was tot 1.12.1974 werkzaam bij het Laboratorium voor Verkeerskunde.



1. Voertuigtrajectoriën.



2. Interieur van de meetwagen, waarin opgesteld de registratie-apparatuur van het VDV-systeem.

tuigen op verschillande raaien te koppelen.
De verkeersdataregistratie en de verkeersdataverwerking vormen te zamen het systeem VERKEERSDATAVERZAMELING (afgekort: VDV-systeem).
In dit artikel worden de deelsystemen verkeersdataregistratie en verkeersdataverwerking in het kort uiteengezet. Voor meer informatie over het VDV-systeem wordt verwezen naar de LVV-memoranda in de referentielijst. Deze memoranda bevatten veelal een meer uitgebreide beschrijving van bepaalde elementen van het systeem.

2. Probleemstelling

Bestuderen van verkeersstromen op micro-niveau vereist informatie over het rijgedrag van de *individuele* voertuigen op een bepaald wegvak. Dit rijgedrag kan weergegeven worden in de vorm van voertuigtrajectoriën die een exacte weergave zijn van de tijd-weg-diagrammen van de voertuigen.
Voor het bepalen van de exacte voertuigtrajectorie van een voertuig is continue informatie nodig over de plaats van dat voertuig in een wegvak (positieinformatie) gecombineerd met de tijdstippen waarop het voertuig op al deze plaatsen aanwezig was (tijdinformatie). Om technische redenen is dit continue meetproces niet realiseerbaar, zodat zowel in de positie als in de tijd bemonsterd moet worden.
Bemonstering in de positie wordt gerealiseerd door in het wegvak op een aantal discrete posities meetraaien aan te brengen. Alleen op deze meetraaien worden de tijdstippen van voertuigaspassages bepaald. Bemonstering in de tijd wordt gerealiseerd door deze meetraaien eenmaal per vaste tijdeenheid (scanperiode) af te tasten. Is een voertuigas gepasseerd dan wordt dit gegeven samen met het bijbehorende tijdstip geregistreerd.

Denkend bijvoorbeeld aan een wegvak van ongeveer 1 km., aan een raaiafstand van 50 m. en een scanperiode van 125 usec. kan deze registratie uitsluitend met zeer geavanceerde apparatuur geschieden. Bij het (off-line) verwerken van de tijd- en positie-informatie tot trajectoriën van individuele voertuigen vormen de enorme hoeveelheden verkeersdata een probleem. Na analyse van de verwerking blijkt deze een sterk routinematig karakter te bezitten, zodat automatische verwerking per computer tot de mogelijkheden behoort.
Indeling van de verwerking in een aantal fasen geeft een logische opbouw van het proces, dat ongeordende tijd- en positieinformatie omzet in geordende voertuigtrajectoriën.

3. Deelsysteem verkeersdata-registratie

In verband met de noodzakelijke mobiliteit van het meetsysteem is de apparatuur van het deelsysteem verkeersdataregistratie, uitgezonderd de detectieapparatuur, ondergebracht in een meetwagen (zie afb. 2). In het schema van afb. 3 zijn de elementen van het deelsysteem verkeersdataregistratie aangegeven.
Op het te bemeten wegvak zijn (maximaal) 127 detectoren aangebracht. De signalen van deze detectoren worden ingevoerd in de meetwagen en via een aanpassingsapparaat toegevoerd aan de detectorsignaalmonitor (DSM). Deze tast eenmaal per scanperiode alle 127 detectorsignalen af en geeft de detector- en tijdinformatie af aan de Raytheon 704 mini-computer, die de informatie bewerkt en wegschrijft naar de magneetbandeenheid.

3.1. Verkeersdetectoren

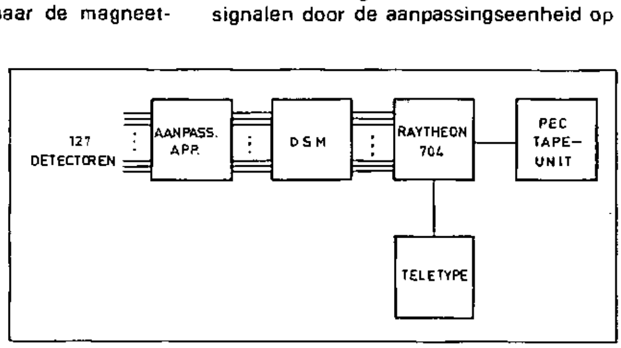
De verwerking van de verkeersdata stelt hoge eisen aan de nauwkeurigheid van de detectoren. Passeert een voertuig een raai dan wordt door een ideale detector op hetzelfde moment een signaal afgegeven. Afhankelijk van het type detector treden echter afwijkingen op in het moment van afgifte van het signaal. Voor een optimale verwerking van de verkeersdata mag deze afwijking niet meer dan 250 usec. bedragen. Deze nauwkeurigheid is met de huidige middelen alleen te bereiken met zogenaamde aspassagedetectoren, die geactiveerd worden door een voertuigwiel. Lusdetectoren zijn niet bruikbaar behalve wanneer het onderzoek de geringe nauwkeurigheid van dit detectortype toelaat. Het meetsysteem kan de signalen van lusdetectoren wel verwerken.

Een raai bestaat per strook uit twee sensoren, die op een meter afstand loodrecht op de rijrichting zijn aangebracht (zie afb. 4). De eerste gedachte ging uit naar coaxsensoren, die een signaal van enkele tientallen millivolts afgeven bij contact tussen voertuigwiel en sensor. Deze coaxsensoren gaven echter afwijkingen die ruim buiten de grens van 250 usec. lagen. Een bruikbaar alternatief vormt de luchtslangsensor, bestaande uit een dunne rubber luchtslang met een buitendiameter van 6 mm, die strak gespannen aangebracht wordt. Aan het ene uiteinde is deze luchtslang afgesloten en aan het andere uiteinde is een transducer aangebracht, die een signaal van enkele millivolts afgeeft op het moment dat een voertuigwiel de slang plat drukt. Dit transducersignaal wordt versterkt en als pulsvormig detectorsignaal ingevoerd in de meetwagen.

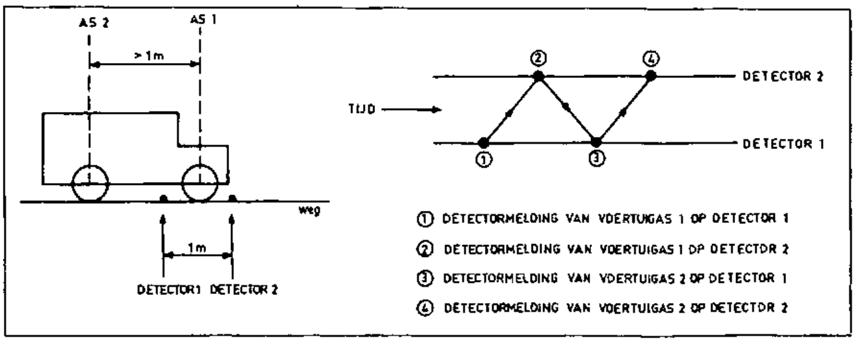
3.2. Detectorsignaalmonitor

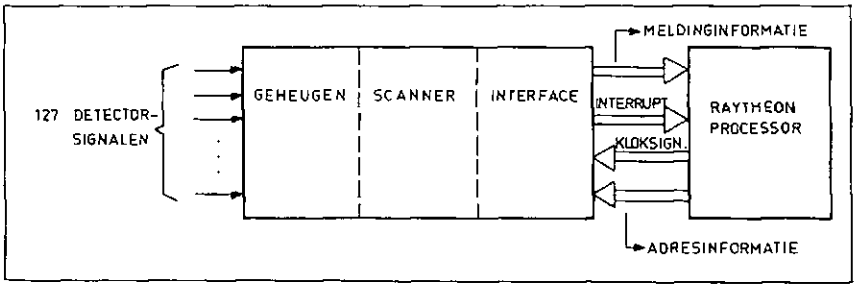
In de meetwagen worden de detectorsignalen door de aanpassingseenheid op

3. Schema apparatuur deelsysteem verkeersdataregistratie.

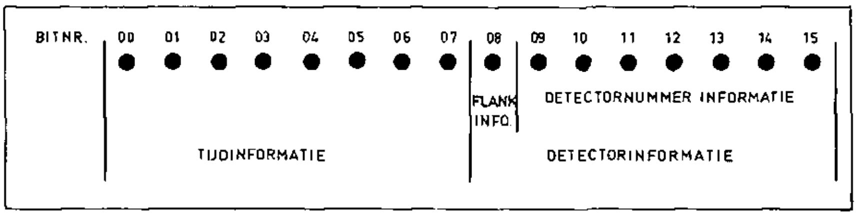


4. Alternering tussen de detectormeldingen van 2 detectoren van een raaistrook.





5. *Blokschema detectorsignaalmonitor.*



6. *Meldinginformatie.*

het juiste spanningsniveau gebracht en toegevoerd aan de detectorsignaalmonitor. De DSM transformeert de signalen van de 127 detectoren om tot informatie die via het datakanaal van de Raytheon 704 computer getransporteerd kan worden. In het blokschema van afb. 5 zijn de volgende elementen te onderscheiden:
1. *geheugengedeelte*: bevat 127 geheugenelementen waarin de logische waarden van de detectorsignalen gedurende één scanperiode vastgehouden worden,
2. *scannergedeelte:* bestuurt de aftasting van de geheugenelementen,
3. *interfacegedeelte*: verzorgt het datatransport tussen DSM en computer.

Is een detector opgekomen (of afgevallen) dan voldoet het betreffende geheugenelement aan de aftastvoorwaarde, zodat sprake is van een detectormelding. Het scannergedeelte tast alle 127 geheugenelementen per scanperiode éénmaal af.
Bij een detectormelding wordt een interrupt aan de computer gegeven, waarna via een interruptroutine de meldinginformatie ingelezen wordt. Deze meldinginformatie is volgens afb. 6 opgebouwd. De tijdinformatie geeft het tijdstip waarop de detectormelding optrad (met de scanperiode als tijdeenheid), de detectornummerinformatie geeft het nummer van de detector en de flankinformatie geeft aan of de detector opgekomen of afgevallen is. Om de 240 scanperiodes wordt een speciale tijdmelding afgegeven die in de computer de hogere orde tijdelementen bijhoudt (5).

3.3. *Computerprogramma*

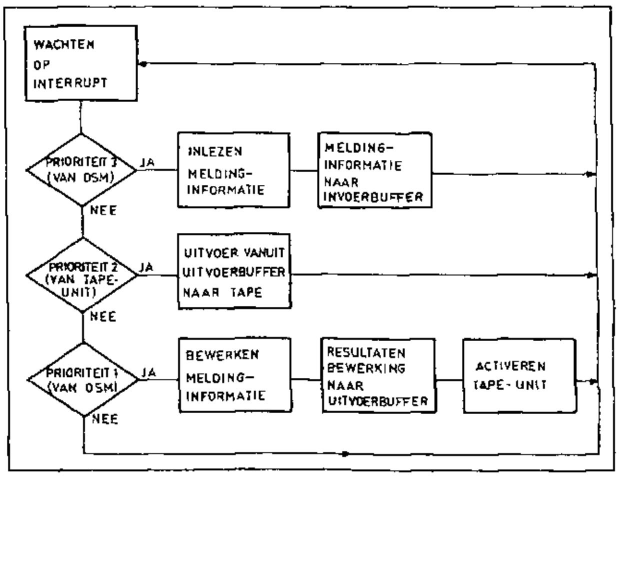
De computer in de meetwagen is een mini-computer type RAYTHEON 704-BASIC configuratie. Op het datakanaal van deze computer is een PEC-magneetband-eenheid aangesloten. De magneetband met de verkeersdata dient als basis voor de off-line verwerking van deze data tot de voertuigtrajectoriën.
Het computerprogramma van het deelsysteem verkeersdataregistratie is geschreven in SYM I-Assembler, en gesplitst in 3 routines die bestuurd door interrupts op basis van prioriteit worden afgewerkt (6).
Ontvangt de computer een interrupt van de DSM dan start een interruptroutine (prioriteit 3) die de meldinginformatie inleest en opslaat in de invoerbuffer. Hierna is deze interruptroutine weer vrij voor het inlezen van de informatie van de volgende melding (zie afb. 7).
Tot het optreden van deze volgende melding bewerkt een andere interruptroutine (prioriteit 1) de meldinginformatie uit de meldbuffer. De resultaten van deze bewerking worden opgeslagen in de uitvoerbuffer waarna de magneetbandeenheid geactiveerd wordt.
Als de magneetbandeenheid informatie kan ontvangen geeft deze een interruptsignaal af dat een interruptroutine (prioriteit 2) start die één byte uit de uitvoerbuffer wegschrijft op magneetband. De prioriteiten die deze interruptroutines t.o.v. elkaar bezitten maken een snelle reactie op interrupts van detectormeldingen (hoogste prioriteit) en op interrupts van de magneetbandeenheid (lagere prioriteit) mogelijk, terwijl in de resterende tijd de meldinginformatie bewerkt wordt (laagste prioriteit). Op deze manier kan het computerprogramma 2 tot 3 detectormeldingen per scanperiode van 125 usec. inlezen. Omdat in het uiterste geval gemiddeld één melding per 4 msec. optreedt is ruim voldoende tijd eanwezig voor de bewerking en de uitvoer naar magneetband.
Bij de bewerking van een detectormelding wordt de 16-bits meldinginformatie gesplitst in twee bytes van 8 bits die voorafgegaan door een codebyte op de magneetband geschreven worden (4). De tijd wordt t.o.v. het begin van de meting in de computer bijgehouden met vier tijdelementen met verschillende orde. Bij wijziging van een tijd wordt de inhoud van het tijdelement voorafgegaan door de bijbehorende code op de magneetband geschreven.

4. Deelsysteem verkeersdataverwerking

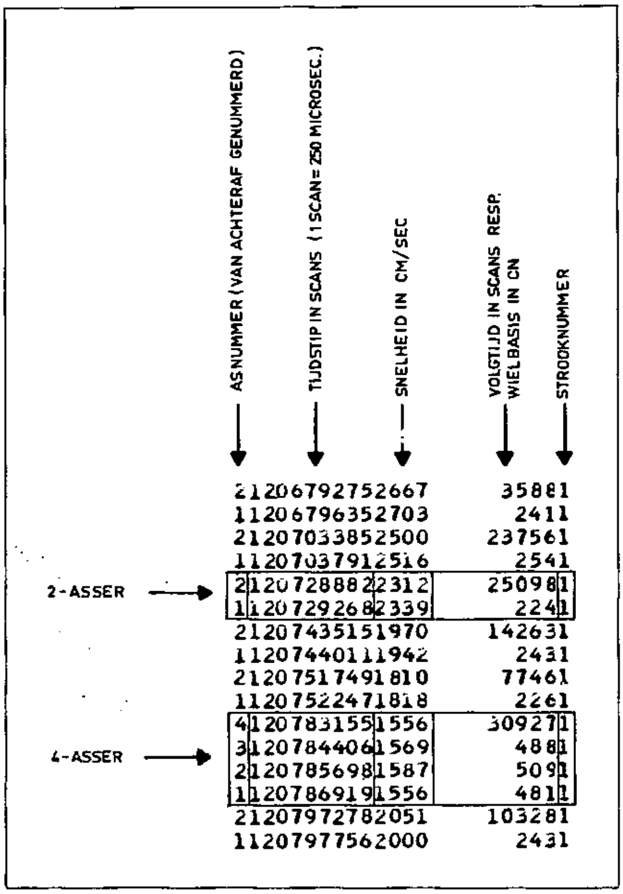
Het vervaardigen van voertuigtrajectoriën uit detectormeldingen laat zich splitsen in het oplossen van drie problemen t.w.:
probleem 1: hoe kunnen uit detectormeldingen *assen* worden verkregen?
probleem 2: hoe kunnen uit assen *voertuigen* worden verkregen?
probleem 3: hoe kunnen met behulp van de voertuigen de *trajectoriën* worden bepaald?
Deze problemen zullen achtereenvolgens behandeld worden.

4.1. *Probleem 1: Het verkrijgen van assen uit detectormeldingen*

Het vormen van assen geschiedt per raaistrook (twee detectoren die op één rijstrook in een raai liggen en gebruikt worden voor het detecteren van aspassages). Op een raaistrook liggen 2 detectoren op een onderlinge afstand van 1 meter (zie afb. 4). Er van uitgaande dat elke wielbasis groter dan 1 meter is, zal elke as eerst detector 1 en vervolgens detector 2 aanraken en zal detector 1 pas weer door de volgende as worden geactiveerd nadat detector 2 is aangeraakt. Op deze wijze ontstaat tussen detector 1 en detector 2 een alternerende reeks detectormeldingen, wat de *alternering* tussen de detectormeldingen van twee detectoren wordt genoemd (zie afb. 4). Een as wordt verkregen door een geregistreerd tijdstip op detector 1 te verbinden met



7. *Stroomschema computerprogramma.*



8. Gedeelte van een strookbestand.

9 (onder). Gedeelte van een moederbestand (gebaseerd op 3 raaien).

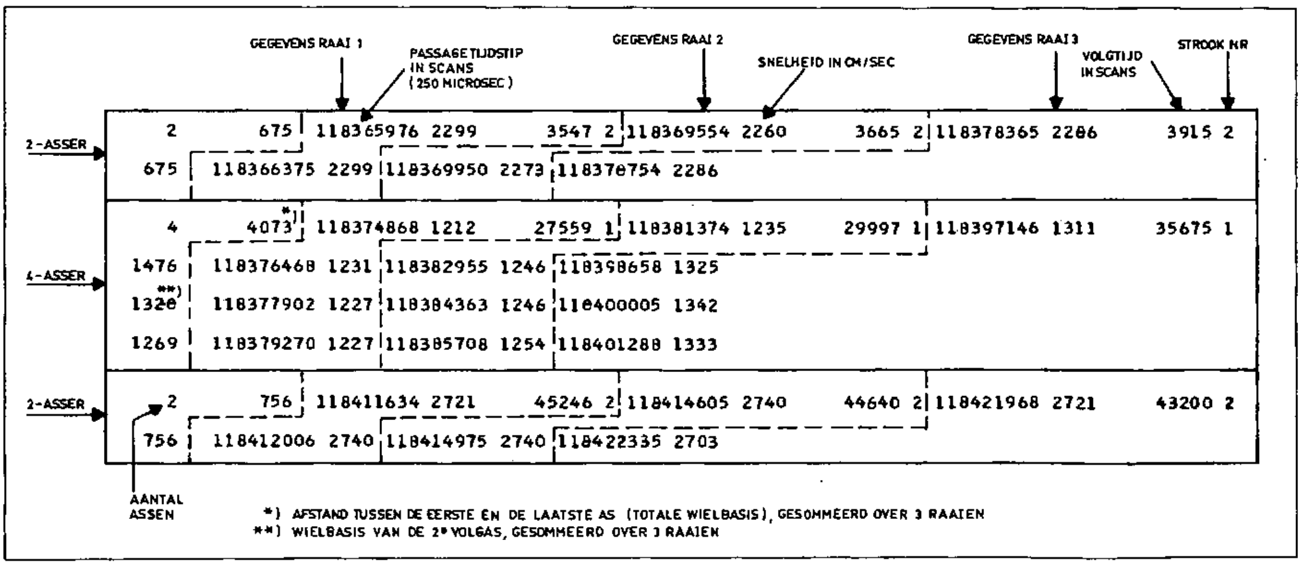
het, in de tijdschaal, eerstvolgende tijdstip op detector 2. De registratieapparatuur werkt niet altijd feilloos. Ook door andere oorzaken kan de elternering worden verstoord zodat detectormeldingen gemist of abusievelijk geregistreerd worden (niet-ontstoorde motoren leveren bijvoorbeeld onder bepaalde omstandigheden een surplus aan detectormeldingen). Aan het programmatuursysteem is de eis gesteld dit soort hardware-fouten te analyseren en, daar waar mogelijk, te corrigeren.

Tijdens het vervaardigen van het systeem was er niets bekend over de frequentie waarmee de verstoring van de alternering zich zou voordoen. Bij de metingen die tot nu toe gehouden zijn, bleek dat bij 95% van de voertuigen die een raaistrook passeerden verstoring van de alternering niet optrad. Zouden in iedere raaistrook de overige 5% voertuigen worden geëlimineerd dan zou het resultaat bij de trajectorievorming (waarbij een voertuig dat op raai 1 gepasseerd is teruggezocht wordt op raai 2, de voertuigen die op raai 1 en op raai 2 gepasseerd zijn worden teruggezocht op raai 3, enz.) afhankelijk zijn van het aantal raaien in het meetvak. Bij 16 raaien bijvoorbeeld zou het aantal voertuigen dat op *iedere* raai correct geregistreerd is gering zijn, n.l. $0.95^{15} \times 100\% \approx 47\%$ van het aantal voertuigen dat het meetvak gepasseerd is. Het is daarom gewenst dat het aantal geëlimineerde voertuigen minimaal is. De heuristische wijze waarop getracht is dit te bewerkstelligen zal bij de behandeling van probleem 2 worden bekeken.

4.2. Probleem 2: Het verkrijgen van voertuigen uit assen

Indien een aantal assen van een raaistrook op een rij in een computergeheugen staat, dan kunnen van de assen de snelheden worden bepaald. Ook kunnen aan de hand van de snelheden de afstanden tussen twee opéénvolgende assen (wielbases) worden bepaald. Een invoerparameter van het systeem is de maximaal toegestane wielbasis. Stel dat deze bijvoorbeeld 7 meter is. Indien er nu wielbases groter dan 7 meter voorkomen dan worden de twee assen, waartussen de wielbasis groter dan 7 meter is, niet aan hetzelfde voertuig toegekend. Er kan tussen de twee assen een scheiding worden aangebracht, een zogenaamde voertuigenscheider. In het geval dat twee voertuigen zo dicht op elkaar rijden dat de afstand tussen de laatste as van voertuig 1 en de eerste as van voertuig 2 kleiner dan of gelijk aan 7 meter is zullen de twee assen onterecht aan één voertuig worden toegekend. Daarom wordt het begrip *groep* geïntroduceerd. Onder een groep wordt verstaan: een aantal assen dat op grond van de maximaal toegestane wielbasis voorlopig aan één voertuig toegekend wordt totdat tijdens de trajectorievorming wordt aangetoond dat één of meer assen aan een ander voertuig toebehoren. De groep wordt dan onmiddellijk gesplitst in meerdere groepen.

Het begrip voertuigenscheider wordt nu vervangen door *groepenscheider*, zijnde het kenmerk dat in een as wordt aangebracht als zeker is dat de as niet tot dezelfde groep behoort als zijn voorgaande as. Het aantal groepen dat meer dan één



voertuig representeert zal sterk afhangen van de onderhavige verkeersstroom. Bij de meting die in punt 5 genoemd wordt bleek dat ruim 97% van de groepen slechts uit één voertuig bestond.
De assen die in het computergeheugen staan zijn op bovengenoemde wijze onderverdeeld in een aantal groepen. Van elke eerste as van een groep kan de berekende wielbasis worden omgezet in een volgtijd. De zo ontstane groepen, die per raaistrook worden bepaald, worden *resultaatgroep* of *strookgroep* genoemd. Strookgroepen zijn groepen die in het *strookbestand* worden opgenomen. Het strookbestand bevat de gegevens van alle groepen die op één raaistrook gedurende de meting passeren (zie afb. 8). Als de alternering niet verstoord is en de assen gevormd worden zoals vermeld in de beschrijving van probleem 1 is het identificeren van de groepen een betrekkelijk eenvoudige zaak. Anders wordt het wanneer de alternering verstoord is. Bij het inlezen van de tijdstippen van de twee detectorbestanden, die de tijdstippen van detector 1 respectievelijk detector 2 van de betreffende raaistrook bevatten, wordt de alternering gecontroleerd. Daar waar de alternering verstoord is worden zogenaamde dummy-tijdstippen gegenereerd. De dummy-tijdstippen krijgen de waarde -1. Er worden zoveel dummy-tijdstippen gegenereerd als nodig is om de alternering te herstellen.
Er kunnen in het computergeheugen dus assen staan die naast één tijdstip één dummy-tijdstip bevatten. Zo'n as wordt een *onvolledige as* genoemd, dit in tegenstelling tot een *volledige as*, die uit twee tijdstippen bestaat. Het aanbrengen van groepenscheiders kan alleen geschieden tussen volledige assen. De ideale situatie om een groepenscheider aan te brengen doet zich voor wanneer van 4 opéénvolgende volledige assen alleen de wielbasis tussen de tweede en derde as groter dan 7 meter is. Tussen de tweede en derde as wordt dan een groepenscheider aangebracht, hetgeen betekent dat de derde as een kenmerk krijgt. In bijzondere gevallen is het mogelijk om een groepenscheider aan te brengen anders dan in de bovengenoemde ideale situatie.
Deze mogelijkheden staan vermeld in (8). Groepen die onvolledige assen bevatten worden nader onderzocht. Daarbij is het mogelijk dat, na heuristische analyse, tijdstippen worden geëlimineerd of dummy-tijdstippen omgezet worden in de meest aannemelijke tijdstippen. Tevens is het mogelijk dat na aanpassing van een groep deze gesplitst wordt in meerdere groepen nl. als er in de aangepaste groep wielbases groter dan 7 meter voorkomen.

4.3. Probleem 3: Het vervaardigen van de voertuigtrajectoriën

Met behulp van het VDV-systeem worden de trajectoriegegevens op de raaien in het meetvak vastgelegd in een zogenaamd *moederbestand* (zie afb. 9). Dit moederbestand bevat *moedergroepen*. Een moedergroep bevat, voor een aantal discrete punten $x^{(i)}$, de trajectoriegegevens van een voertuig dat het meetvak tijdens een meting gepasseerd is.
De punten $x^{(i)}$ stemmen overeen met de raaien in het meetvak. De afstanden tussen de raaien (raaiafstand) mogen tussen ieder tweetal opéénvolgende raaien verschillend zijn.
Voor ieder voertuig wordt per raai in het moederbestand vastgelegd:
1. passagetijdstip;
2. snelheid;
3. volgtijd;
4. strooknummer in de raai.

Om een moederbestand te verkrijgen moet elke groep die gepasseerd is op raai 1 teruggezocht worden op de raaien 2 tot en met n. Eerst worden de groepen van raai 1 *gekoppeld* aan een strookgroep op raai 2. Het resultaat is een moederbestand gebaseerd op 2 raaien. Vervolgens wordt nu getracht de groepen uit dit moederbestand te koppelen aan een strookgroep behorende tot raai 3, enz. Het eindresultaat wordt een op n (= aantal raaien in het meetvak) raaien gebaseerd moederbestand.
In een raai zullen meestal meerdere rijstroken gelegen zijn. De vraag is nu in welke volgorde de rijstroken moeten worden afgezocht omdat o.a. rekening moet worden gehouden met het wisselen van rijstrook. Ook is in het algemeen de geometrie van de weg niet overal op het meetvak hetzelfde (het aantal rijstroken kan bijvoorbeeld variëren) en kan het verkeersgedrag in verschillende raaivakken duidelijk verschillen (o.a. bij invoeg- en uitrijpunten). Om voorgaande redenen zou het wenselijk zijn om het desbetreffende programma te voorzien van een *prioriteitenrelatie* tussen de rijstroken van raai j en de rijstroken van raai j+1. De mogelijkheid hiertoe wordt geboden door een zogenaamde relatiematrix te definiëren en deze via ponskaarten aan het programma mee te geven. Het voordeel van deze methode is dat voor elke rijstrook op raai j een andere prioriteitenlijst voor het in volgorde afzoeken van rijstroken op raai j+1 kan worden opgegeven. Verwacht wordt dat daardoor de totale zoektijd, benodigd om alle groepen te koppelen, gunstig zal worden beïnvloed.
In het voorgaande is gesproken over het koppelen van een groep op raai j met een strookgroep op raai j + 1. Via een zoekalgorithme, dat o.a. gebruik maakt van de gegevens van de zojuist genoemde relatiematrix, komt de één of andere strookgroep in aanmerking om met de onderhavige groep uit het moederbestand (moedergroep) gekoppeld te worden.
De beslissing van al dan niet koppelen hangt af van het al dan niet voldoen aan de zogenaamde *koppelingscriteria*. Er worden twee koppelingscriteria gebruikt:
1. Een criterium voor een maximaal toegestane afwijking met betrekking tot de *wielbasis*. Dat wil zeggen dat iedere wielbasis (tussen 2 opéénvolgende assen) van de moedergroep niet meer dan een bepaalde afwijking mag hebben ten opzichte van de overeenkomstige wielbasis van de in aanmerking komende strookgroep. De maximaal toegestane afwijking wordt in het programma, dat de koppeling van raai j naar raai j+1 tracht te bewerkstelligen, als parameter ingevoerd en wordt δ genoemd. Indien bijvoorbeeld een moedergroep bestaat uit 3 assen, met wielbases WM12 en WM23 wordt gekoppeld aan een strookgroep die eveneens bestaat uit 3 assen, met wielbases WS12 en WS23 dan moet dus gelden:

   $|WM12-WS12| \leqslant \delta$ én $|WM23-WS23| \leqslant \delta$

2. Een criterium voor de maximaal toegestane afwijking met betrekking tot de *snelheid*. Het criterium luidt:

$$\left| \frac{1}{2}(v_1 + v_2) - \frac{\text{raaiafstand}}{T_2 - T_1} \right| \leqslant \varepsilon$$

met:
v_1 = snelheid eerste as moedergroep op raai j
v_2 = snelheid eerste as strookgroep op raai j+1
T_1 = passagetijdstip eerste as moedergroep op raai j
T_2 = passagetijdstip eerste as strookgroep op raai j+1
ε = maximaal toegestane afwijking, die als parameter aan het betreffende programma wordt meegegeven.

In woorden betekent dit criterium dat de gemiddelde snelheid over een raaivak niet meer dan ε mag verschillen van het gemiddelde van de plaatselijke snelheden op de begrenzende raaien (spot-speeds). De bovengenoemde koppelingscriteria, die gehanteerd worden bij metingen op „rechte" wegvakken, zijn als een eenvoudig verwisselbare eenheid in het betreffende programma opgenomen. De eigenschap „eenvoudig verwisselbaar" is noodzakelijk omdat bij ieder type meting de koppelingscriteria anders kunnen zijn.

4.4. Organisatie van het systeem

Bij de ontwikkeling van het programmatuursysteem waren de volgende punten van belang:
1. Het meetsysteem is bestemd voor metingen op *„rechte" wegvakken*.
2. Het programmatuursysteem dient *„hardware-fouten"*, zoals bijvoorbeeld gemiste of abusievelijk geregistreerde detectormeldingen, te analyseren en, daar waar mogelijk, te corrigeren.
3. De programma's moeten door belangstellenden gebruikt kunnen worden. Bij de keuze van de programmeertaal is dit van belang.

Het systeem VERKEERSDATAVERZAMELING bestaat uit de volgende delen:

Fase 0 Registratie	data registratie	VDV systeem
Fase 1: Decodering Fase 2: Groepsidentificatie Fase 3. Trajectorievorming Fase 4: Utilities	data verwerking	

In fase 0 worden de detectormeldingen in gecodeerde vorm op een magneetband gezet. In de code worden het detectornummer en het tijdstip van de melding opgenomen.
Deze gegevens worden in fase 1 verwerkt. Dit verwerken betekent dat de gecodeerde gegevens omgevormd worden tot tijdstippen uitgedrukt in eenheden van 250 mi-

croseconden (scans). Verder worden de gegevens gesplitst in max. 127 uitvoerfiles (detectorbestanden) en wel zodanig dat iedere uitvoerfile de tijdstippen van één detector bevat.
Fase 2 verzorgt de groepsidentificatie. Zowel het vormen van assen uit detectormeldingen (probleem 1) als het vormen van groepen uit assen (probleem 2) wordt in dit gedeelte van het programmatuursysteem gerealiseerd.
In fase 3 worden de voertuigtrajectoriën vervaardigd (probleem 3).
Fase 4 bestaat uit het hulpprogramma VD99. Dit programma biedt de gebruiker van het systeem VERKEERSDATAVERZAMELING een aantal mogelijkheden om te kunnen manipuleren met tijdenbestanden, strookbestanden en moederbestanden.

Zo is het mogelijk om bestanden te copiëren, te ponsen, af te drukken, te mengen, te wijzigen, deelbestanden te construeren, enz. Het programma bezit 22 opties. Iedere optie kan worden geactiveerd door een opdrachtkaart en een aantal andere stuurkaarten, afhankelijk van de te activeren optie. Het aantal opdrachtkaarten is vrij, zodat het mogelijk is meerdere bestanden en/of meerdere opties in één run sequentieel te activeren. In het algemeen wordt niet meer naar een vorige fase teruggegaan indien de verwerking tot in een bepaalde fase gevorderd is. Voor de fasen 2,3 en 4 is als programmeertaal FORTRAN-IV gekozen. Deze taal is op praktisch iedere computerconfiguratie geïmplementeerd. De programma's uit deze fasen bestaan te zamen uit ruim 4000-FORTRAN-statements (exclusief standaard-programmatuur en comment-statements). In het kader van dit artikel is het niet mogelijk alle programma's en de toegepaste heuristische werkwijze te beschrijven. Wel zijn flow-charts opgenomen (zie afb. 10 en 11) om een indruk te geven. Gedetailleerde beschrijvingen zijn te vinden in de memoranda welke in de referentielijst zijn opgenomen.

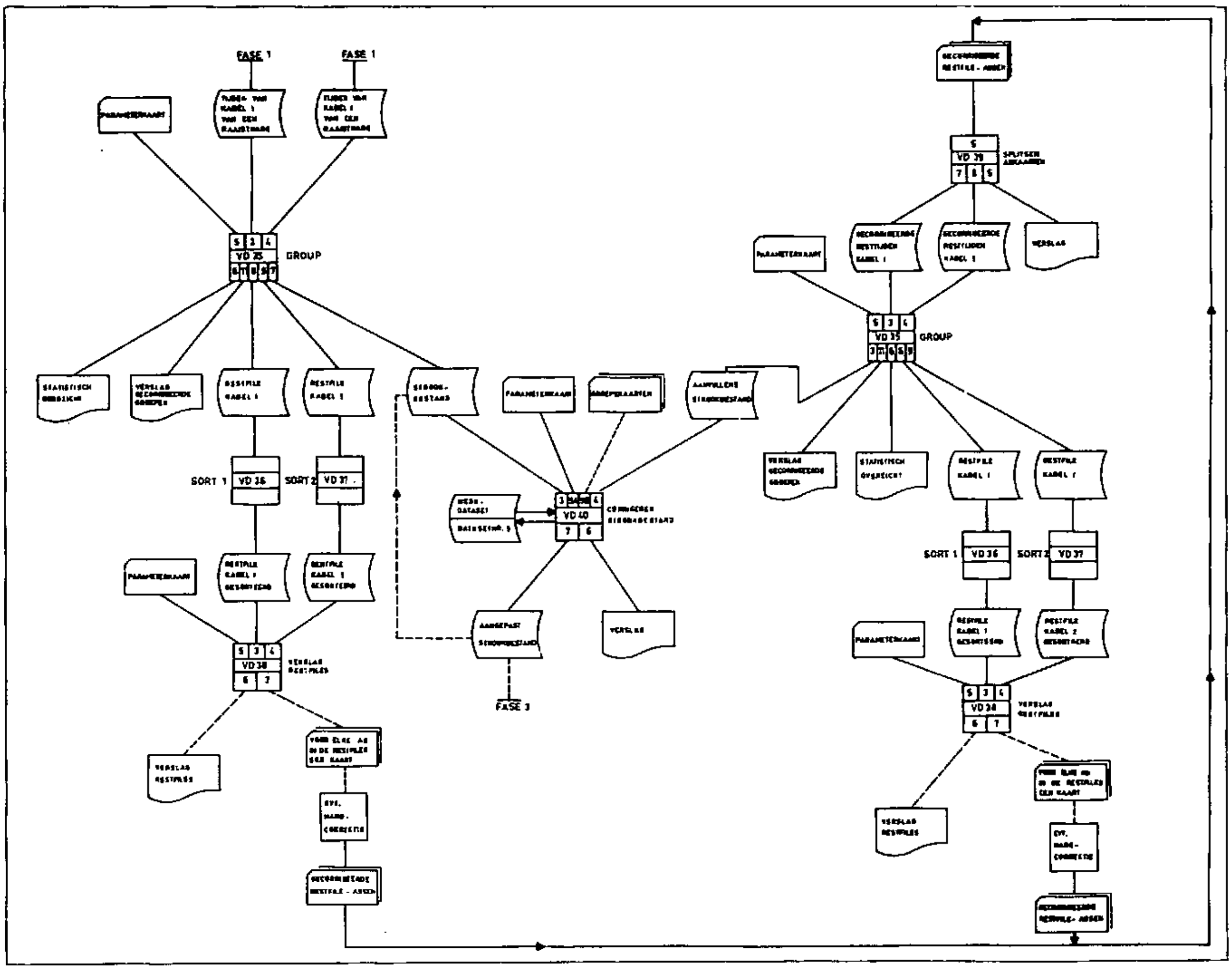
TABEL 1. Resultaten meting Kruithuisweg voor trajectorievorming, fase 3

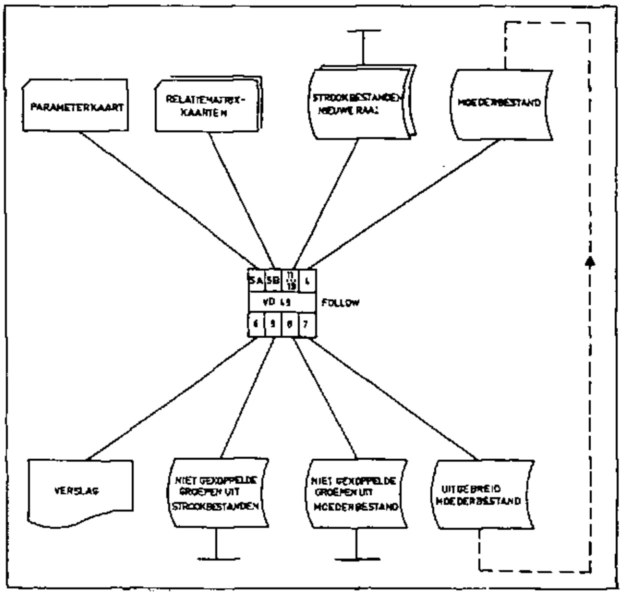
aantal raaien	aantal voertuigen	aantal afvallers	perc. gekoppeld	perc. gekoppeld totaal	rekentijd IBM 370/158	totale rekentijd
1	850					
1+2	841	9	98.9	98.9	16 sec.	16 sec.
1+2+3	835	6	99.3	98.2	21 sec.	37 sec.
1+2+3+4	830	5	99.4	97.6	25 sec.	62 sec.

5. Resultaat van een verrichte meting

Op de provinciale weg no. 53 (Kruithuisweg) te Delft is op de zuidoostelijke rijbaan ter hoogte van de Schieweg een meting gehouden. Doordat de detectoren (in dit geval coaxiaalkabels) nog niet nauwkeurig genoeg waren zijn de invoerparameters die de maximaal toegestane afwijkingen in criterium W en criterium V aangeven, ruim genomen. Voor δ is 25 cm en voor ε is 200 cm/sec genomen. Ondanks deze ruime tolerantie met betrekking tot de afwijkingen in de koppelingscriteria zijn de resultaten hoopgevend. De verkeersomstandigheden in het meetvak waren:

10. Flow chart verkeersdataverzameling - groepsidentificatie.





11. Flow chart verkeersdataverzameling - trajectorievorming.

a. éénrichtingsbaan;
b. twee rijstroken;
c. vier raaien;
d. ongeveer 1000 voertuigen per uur;
e. ongeveer 5% strookwisselaars.

Voor de trajectorievorming, fase 3, werd het resultaat gevonden als aangegeven in tabel 1.

De gehele meting was tevens met een videorecorder op videotape vastgelegd. Na controle bleek dat alle door het programmatuursysteem vervaardigde trajectoriën, ook bij strookwisselaars, overeenstemden met de werkelijkheid. De „uitvallers" werden afgewezen op grond van het wielbasiscriterium. Dit was te wijten aan de soms plotselinge en tijdelijke onnauwkeurigheden van de detectoren.

6. Slotopmerking

Het systeem VERKEERSDATAVERZAMELING is ontwikkeld voor gebruik door het Laboratorium Voor Verkeerskunde (L.V.V.), doch ook andere onderzoekinstellingen kunnen in de gelegenheid worden gesteld het VDV-systeem te gebruiken. Voor het L.V.V. geschiedt de dataverwerking met behulp van de IBM 370/158 computer van de TH-Delft. Afbeelding 12 geeft een overzicht van de verwerking van de detectormeldingen tot voertuigtrajectoriën.

Referenties

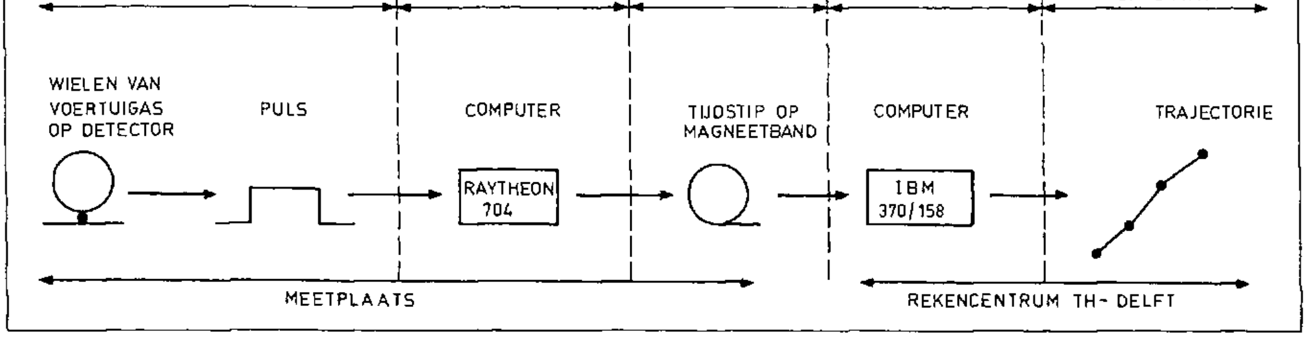
1. Makigami, Y. et al, Three-Dimensional Representation of Traffic Flow, Transportation Science Vol. 5 (1971), p. 302.
2. Volmuller, J.,Verkeersstromentheorie, collegedictaat e12., Delft, Technische Hogeschool, Afdeling der Weg- en Waterbouwkunde, 1973, p. 14.
3. Jansen, G.R.M. — Bovy, P. H. L., Verkeers- en vervoersresearch aan de TH Delft 1970. Verkeerstechniek Vol.22(1971), p. 119.
4. Mozes, D., Registratie van de meldinginformatie op magneetband. Delft, THD-LVV, mei 1973, memorandum No. VDR/2/73.2, 28 pag.
5. Mozes, D., Detektorsignaalmonitor met detektorinformatie als binair detectornummer. Delft THD-LVV, mei 1974, memorandum No. VDR/1/74.1, 93 pag.
6. Mozes, D., VDR-software. Computerprogramma's voor verkeersdataregistratie. Delft, THD-LVV, okt. 1974, memorandum No. VDR/1/74.2, 108 pag.
7. Dijk, A.J. van, Verkeersdataverzameling. Fase 2: Samenstellen van groepen. Onderdeel: Voorstel tot behandeling van potentiële groepen. Delft, THD-LVV, juli 1972, memorandum No. VDV/1/72.5, 70 pag.
8. Dijk, A.J. van, Verkeersdataverzameling. Fase 2: Groepsidentificatie, Versie 0. Delft, THD-LVV, mei 1973, memorandum No. VDV/2/73.1, 88 pag.
9. Dijk, A.J. van, Verkeersdataverzameling. Fase 2: Groepsidentificatie, Versie 0,Programma's. Delft, THD-LVV, mei 1973, memorandum No. VDV/2/73.2, 58 pag.
10. Dijk, A.J. van, Verkeersdataverzameling. Hulpprogramma's testfase. Programma's VDH1 t/m VDH9. Delft, THD-LVV, mei 1973, memorandum No. VDV/2/73.3, 39 pag.
11. Dijk A.J. van, Verkeersdataverzameling. Fase 2: Proefgevallen. Delft, THD-LVV, mei 1973, memorandum No. VDV/2/73.4, 40 pag.
12. Dijk, A.J. van, Verkeersdataverzameling. Fase 3: Trajectorievorming. Programma: FOLLOW (VD45). Delft THD-LVV, dec. 1973, memorandum No. VDV/2/73.9, 63 pag.
13. Dijk, A.J. van, Verkeersdataverzameling. Fase 3: Trajectorievorming, Versie 0, Programma's. Delft, THD-LVV, dec. 1973, memorandum No. VDV/2/73.10, 30 pag.
14. Dijk, A.J. van, Verkeersdataverzameling. Fase 4: Utilities, Versie 0. Delft, THD-LVV, jan. 1974, memorandum No. VDV/2/74.1, 72 pag.
15. Dijk, A.J. van, Verkeersdataverzameling. Fase 4: Utilities, Versie 0, Programma's. Delft, THD-LVV, jan. 1974, memorandum No. VDV/2/74.2, 52 pag.
16. Dijk, A.J. van, Verkeersdataverzameling. Fase 4: Utilities, Proefgevallen. Delft, THD-LVV, dec. 1974, memorandum No. VDV/2/74.5, 61 pag.
17. Dijk, A. J. van, Verkeersdataverzameling. Een computertoepassing in de verkeerstechniek. Delft, THD-LVV, dec. 1973, memorandum No. VDV/2/73.14, 48 pag.

Summary

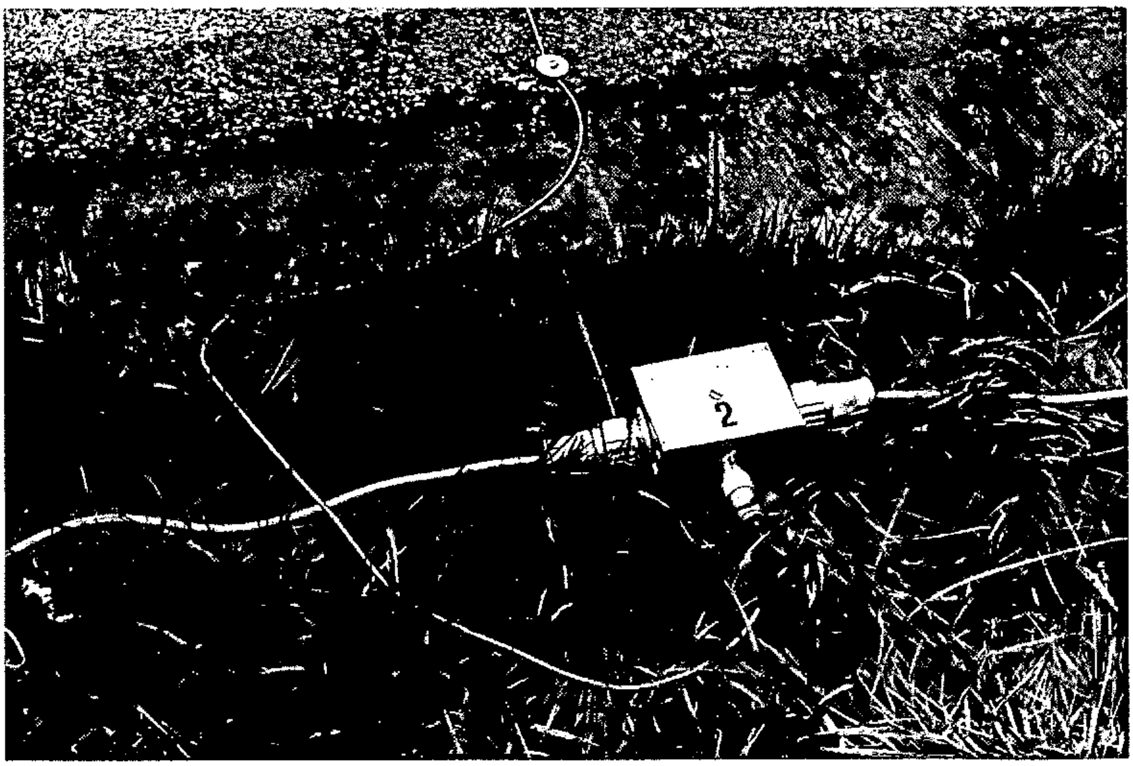
Traffic data collection

In behalf of the research of traffic streams it is desirable to collect data about traffic streams.
The Transportation Research Laboratory from the Department of Civil Engineering of the University of Technology in Delft, has the equipment to carry out traffic measurements.
With this equipment and with the help of traffic detectors the points of time of the activation of the detectors by the vehicle axes are recorded on magnetic tape.
This article describes the equipment and a software system which derives vehicle trajectories from the detector records on the magnetic tape.

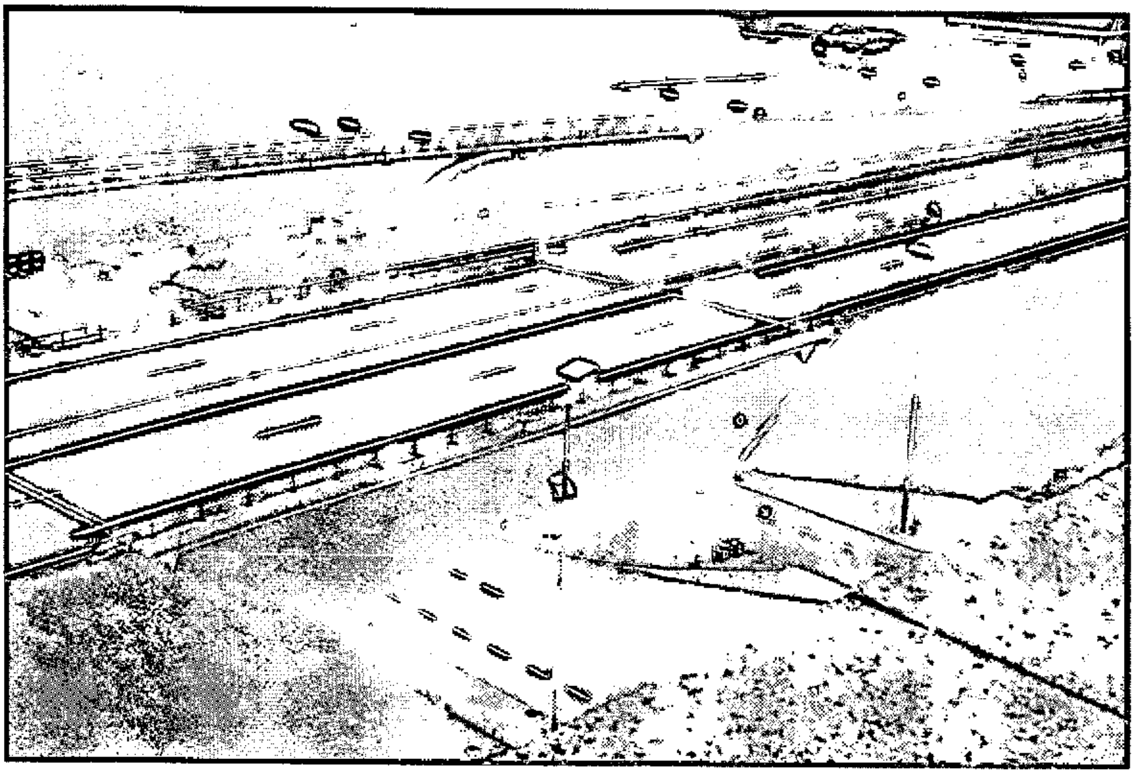
12. Overzicht van de verwerking van de detectormeldingen tot voertuigtrajectoriën.



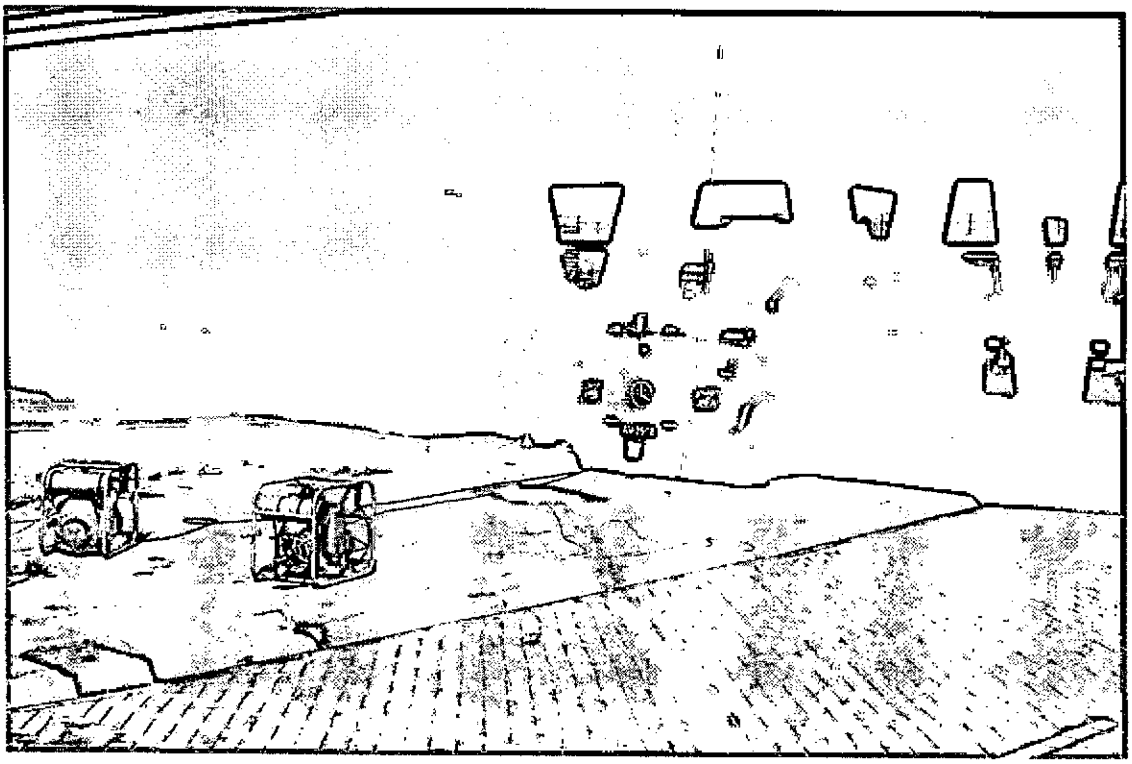
Interieur van de meetwagen, waarin opgesteld de registratie-apparatuur van het VDV-systeem



Coaxiaalkabel met versterker



Overzichtsfoto meetplaats Kruithuisweg Delft
Het meetvak (onderste rijbaan) begint bij de voeg van het viaduct, links onder



De meetwagen was onder het viaduct geplaatst

Publication 14

BIBLIOSYSTEM

Computerhulp bij het samenstellen van

bibliografieën

Composing bibliographies by means of a computer

1971

Adding more people to a late software project
makes it more costly, but not necessarily later.

Tarek Abdel-Hamid & Stuart Madnick [1991]

This article previously appeared in:

Infarmatie, Octaber 1971, pages 448-455

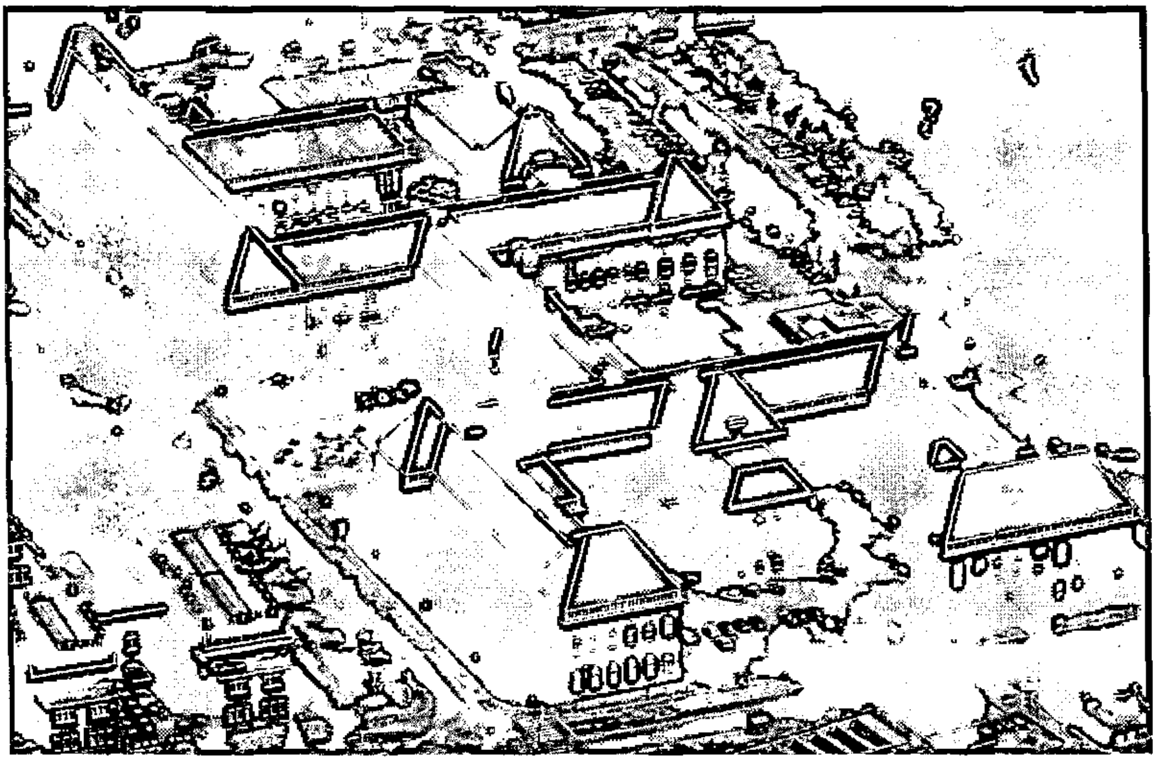
Abstract

The composition of bibliographies and literature lists is part of virtually every researcb project of some significance. In this, a large flexibility with regard to updating and classifying proves to be of great value. In order to meet these demands, the Transportation Research Laboratory of the DUT developed the 'BIBLIOSYSTEM' information system. This article describes the extensive functionalities of 'BIBLIOSYSTEM'. As a result of this article, over 100 businesses and organisations procured the documentation/sources of 'BIBLIOSYSTEM'.

TU Delft

Technische Universiteit Delft

Rekencentrum

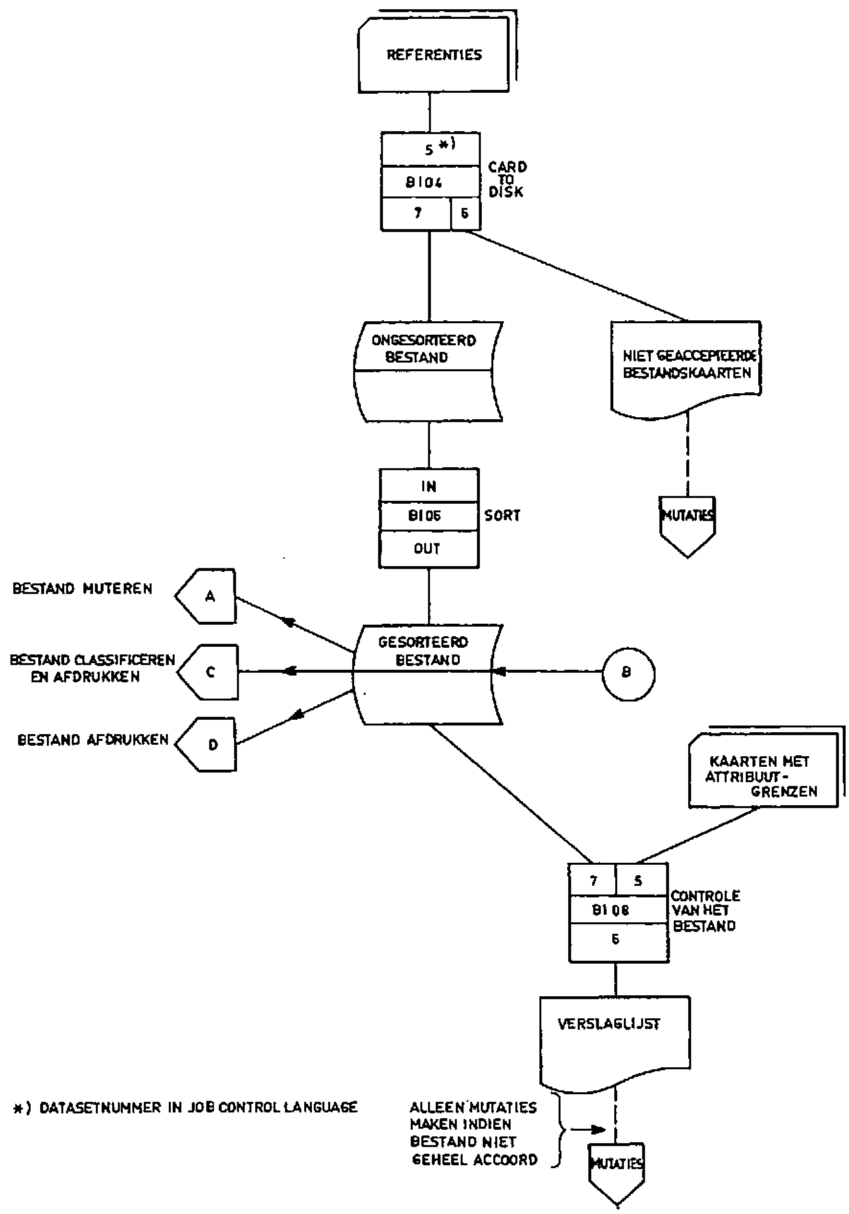


BIBLIOSYSTEM
Computerhulp bij het samenstellen van bibliografieën

door A. J. van Dijk

1 HET KADER, DOEL EN RELEVANTIE VAN HET SYSTEEM

De snel groeiende stroom van publicaties leidt ertoe dat het verzamelen, ordenen en bestuderen van literatuur een steeds belangrijkere plaats in het onderzoek gaat innemen. Het samenstellen van bibliografieën en literatuurlijsten is dan ook onderdeel van vrijwel ieder researchproject van enige betekenis. Een grote flexibiliteit in het bijwerken en classificeren blijkt daarbij van uitermate grote waarde. Deze bewerkingen zijn dermate arbeidsintensief dat zij een gedegen literatuurstudie, vooral wanneer deze van enige omvang is, bemoeilijken.
De klassieke wijze van rapporteren van een literatuurstudie is zeer tijdrovend vanwege de omvang van het typewerk: de onderzoeker houdt een kaartsysteem bij, hij selecteert de in aanmerking komende referenties en laat deze, nadat in de meeste gevallen een classificatie en volgorde is aangebracht, typen.
Zoals in een groeiend aantal gevallen, vormt ook hier dit typewerk een ware bottle-neck. De gevoeligheid voor fouten is zeer groot, waardoor een consciëntieuze controle zeer noodzakelijk is.
Behalve de invloed, die deze omstandigheid heeft op de bewerkingstijd, onderdrukt zij de noodzakelijke flexibiliteit voor wat betreft het bijwerken, maken van diverse classificaties en het rapporteren van de literatuurstudie. De ernst van dit bezwaar moet niet onderschat



Figuur 1a

worden omdat de kwaliteit van de literatuurstudie een zeer grote invloed heeft op de resultaten van het onderzoekwerk.

Al deze factoren leidden tot het idee om het samenstellen van bibliografieën verregaand te automatiseren.

Het is belangrijk er op te wijzen dat het *niet* in de bedoeling lag een volledig 'information-retrieval'-systeem op te bouwen. Deze systemen zijn reeds elders geprogrammeerd en bovendien zijn ze bedoeld voor omvangrijkere vraagstukken hetgeen met zich meebrengt dat zij voor ons probleem te gecompliceerd zijn.

Aan het te ontwikkelen systeem (voortaan 'BIBLIO-SYSTEM' genoemd) werden de volgende eisen gesteld:

1 Het systeem dient flexibel te zijn met betrekking tot bijwerken van het referentiebestand (toevoegen, verwijderen en vervangen van referenties).
2 De referenties moeten in een bepaalde volgorde gebracht kunnen worden.
3 De referenties moeten op een flexibele wijze geclassificeerd kunnen worden.
4 Zoveel mogelijk controles op de gecodeerde referenties behoren door het systeem, op een doelmatige wijze, uitgevoerd te worden.
5 De referenties dienen van een 'label' voorzien te kunnen worden teneinde in een rapport daaraan te kunnen refereren.
6 De lay-out dient presentabel te zijn en op eenvoudige wijze aan te passen aan de (redelijke) eisen van de gebruiker.
7 Het systeem dient zoveel mogelijk onafhankelijk te zijn van bibliografische afspraken omtrent de wijze waarop referenties worden samengesteld.
8 Naast de referenties dient het systeem eveneens de begeleidende informatie te kunnen verwerken.
9 De programma's moeten door zoveel mogelijk belangstellenden gebruikt kunnen worden. Bij de

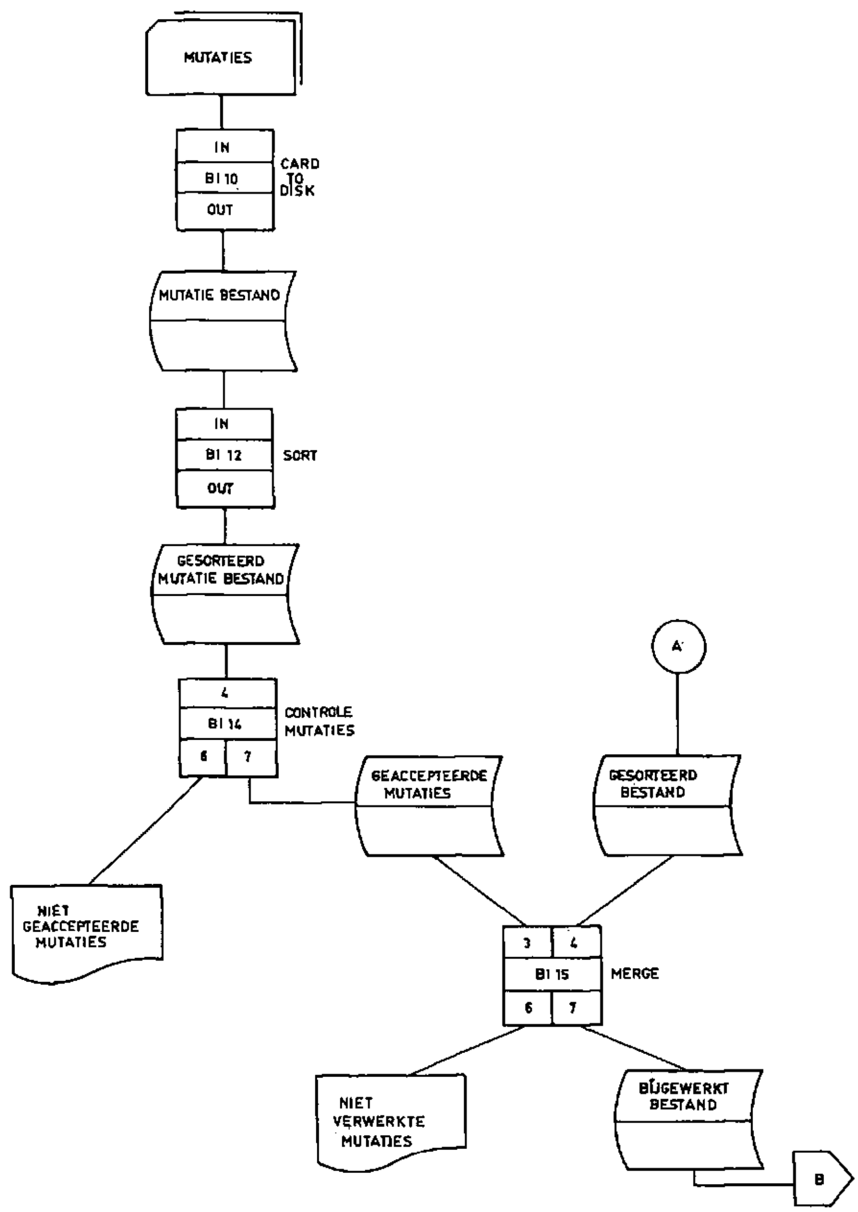


Fig. 1b

keuze van de programmeertaal is deze eis van belang.

Het systeem is ontwikkeld voor eigen gebruik, doch de indruk bestaat dat zeer veel onderzoekers met voordeel het systeem zouden kunnen gebruiken.
Verder is het systeem geschikt om literatuurbezit van researchgroepen, kleine bibliotheken te classificeren en in een catalogus weer te geven. In het laatste geval kan het systeem ook dienen als een eenvoudig 'information system'.

2 BESCHRIJVING VAN HET BESTAND

2.1 Het bestand

Een centrale plaats neemt het *referentiebestand* in, aangezien dit direct of indirect het onderwerp is van alle bewerkingen die door 'BIBLIOSYSTEM' worden uitgevoerd. Het is opgebouwd uit referenties met daarnaast identificatie en informatie die op de verzameling referenties betrekking hebben. De samenstelling van het bestand is aangegeven in figuur 2a.
In bibliografisch opzicht valt een referentie uiteen in een aantal elementen met elk een specifieke functie, te weten gedeelten voor de beschrijving van respectievelijk de auteur, de titel, de locatie en de samenvatting (abstract). Dit vormde een uitgangspunt van de programmering van het systeem. Voor het overige zijn de bibliografische regels volgens welke men de referenties samenstelt niet of nauwelijks van belang bij gebruik van 't systeem.

2.2 De referenties

2.2.1 *Algemeen*

Ter verwerking van de referenties is aan de onder 1. genoemde elementen een specificatiegedeelte toegevoegd.

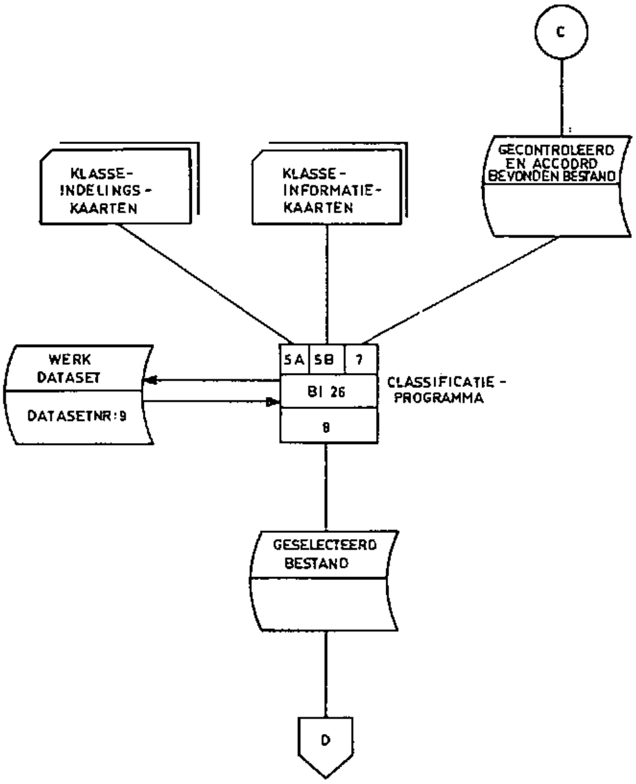


Fig. 1c

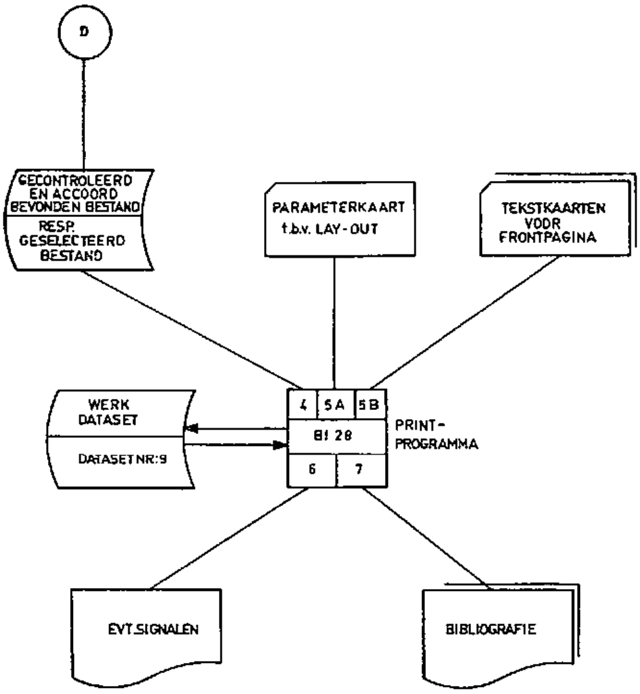


Fig. 1d

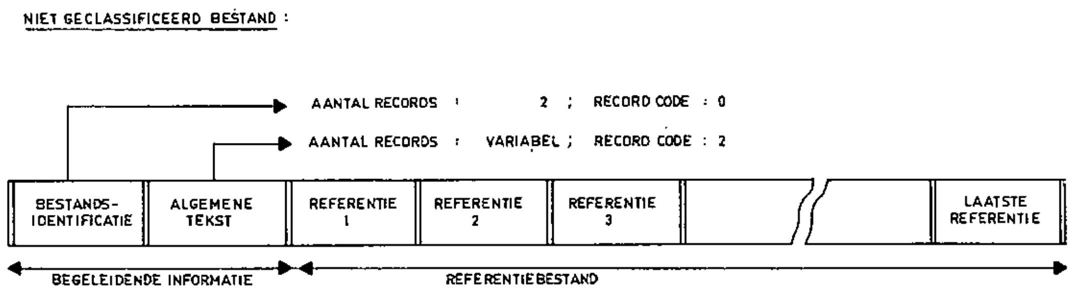
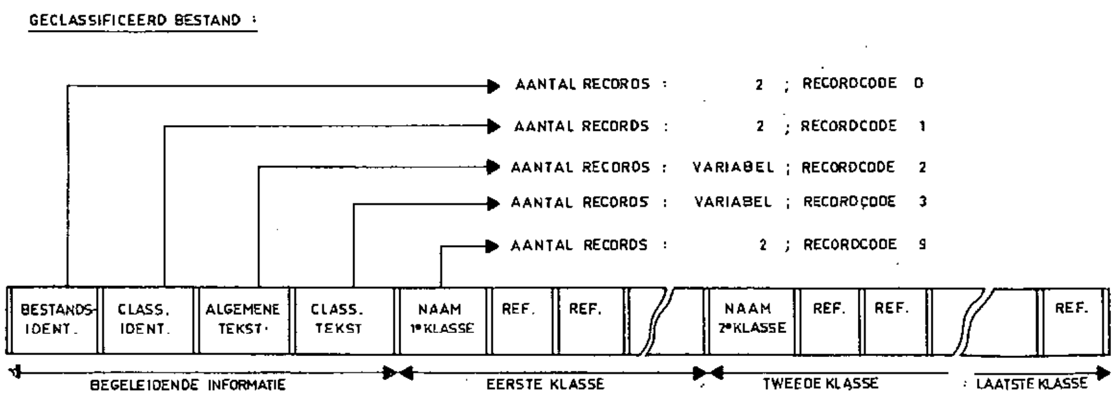


Fig. 2a

Fig. 2b



Een referentie is in programmatisch opzicht dus opgebouwd uit kaarten van de volgende soorten:
1 specificatiekaart,
2 auteurskaart,
3 titelkaart(-en),
4 locatiekaart(-en),
5 abstractkaart(-en).

Gedurende de behandeling van een correct gecodeerde referentie door het systeem blijven de kaarten van één referentie bij elkander en bovengenoemde volgorde blijft gehandhaafd.
De kaarten van de referenties worden verwerkt op basis van een nummer dat eraan wordt toegekend. Dit zg. *bestandsnummer* is op een zodanige wijze samengesteld dat een gemakkelijke verwerking en controle mogelijk is.
Het bestandsnummer is opgebouwd uit achtereenvolgens het zg. *referentienummer*, een code voor de betrokken kaartsoort en een volgnummer voor de kaarten van eenzelfde kaartsoort binnen een referentie.
Het referentienummer is voor iedere referentie van het bestand verschillend.
Het is opgebouwd uit:
- een numerieke code.
  Hoewel niet strikt noodzakelijk, wordt geadviseerd deze code te baseren op de naam van de auteur van de publicatie. De regel die hierbij dan van toepassing is, is dat de rangschikking van de referenties volgens de naam van de eerste auteur in alfabetische zin overeenkomt met die volgens de numerieke code naar opklimmende grootte.
  De transformatie van auteursnamen in numerieke codes geschiedt op gemakkelijke wijze met gebruikmaking van bestaande, daartoe samengestelde tabellen (zie figuur 3).
- de laatste twee posities van het jaartal waarin de publicatie is verschenen.
- een volgnummer.
  Dit volgnummer dient ter onderscheiding van publicaties van één auteur, welke in één jaar verschenen.

Een en ander heeft tot gevolg dat de referenties worden gesorteerd in hiërarchische volgorde op: de naam van de auteur (in alfabetische zin), het jaar van publicatie en het toegekende volgnummer.
Tevens wordt daarmee een voor iedere kaart van het bestand verschillend bestandsnummer verkregen. Dit heeft tot gevolg dat sortering volgens dit nummer van de afzonderlijke kaarten leidt tot een aaneengesloten plaatsing van alle kaarten van één referentie in de reeds genoemde volgorde.

2.2.2 *De specificatiekaart*
Deze kaart bevat enerzijds informatie op basis waarvan de referentie geclassificeerd kan worden en anderzijds informatie ter verwerking van de referentie door de diverse programma's van het systeem.
De kenmerken van de referentie, waarop classificaties kunnen worden gebaseerd, worden vermeld in numerieke vorm in tien attributen. Voorts is er een rubriek ter opneming van het jaartal van publicatie, eventueel met suffix. Dit maakt een eenduidige verwijzing naar de referentie mogelijk (zie 3.5).

MICRO NUMMER	ACHTERNAAM	VOOR-VOEG-SELS	I^e VOORNAAM	2^e VOOR-LETTER
8056	STELTMAN			
8057	STEMLER			
8058	STENEKES			
8059	STENVERS			
8060	STERCKEL			
8061	STERK		ANNA	
8062	STERK		J	
8063	STERK		S	
8064	STERKEN		K	
8065	STERKENBURGH			
8066	STERNVELD			
8067	STERRENBERG			
8068	STET			
8069	STEUR			
8070	STEUR		M	
8071	STEUTELINGS			
8072	STEVENS			
8073	STEVENS		CORNELIS	
8074	STEVENS		H	
8075	STEVENS		JANNA	
8076	STEVENS		M	
8077	STEVENS		PETRUS	
8078	STEVENSE		M	
8079	STEYGERWALT			
8080	STEYN		K	
8081	STEYN	V	R	
8082	STEYVERS			
8083	STICK			
8084	STIEMSMA			
8085	STIENSTRA			
8086	STIEMSTRA		S	
8087	STIGT	V	H	
8088	STIGTER		K	
8089	STIKKELORUM			
8090	STILKENBOOM			
8091	STINKENS			
8092	STIPHOUT			
8093	STIPHOUT	V	JOHANNES	
8094	STIPRIAAN	V		
8095	STOBBELAAR			
8096	STOCKMANN		M	
8097	STOEL		C	
8098	STOEL	VD		
8099	STOELMAN			
8100	STOEP	VD	N	

Fig. 3

2.2.3 *De auteurskaart*
Op deze kaart wordt de auteur van de publicatie vermeld.

2.2.4 *De titelkaart(-en)*
Op deze kaart(-en) wordt de titel van de publicatie vermeld.

2.2.5 *De locatiekaart(-en)*
Deze kaart(-en) bevat(-ten) de informatie die noodzakelijk is ter opsporing van de publicatie, zoals b.v. titel tijdschrift, jaargang, nummer etc.

2.2.6 *De abstractkaart(-en)*
In beginsel zijn deze kaarten bedoeld ter opneming van een samenvatting (abstract) van de referentie. Vanzelfsprekend kan men op deze kaarten, die ook mogen ontbreken, andersoortige informatie (b.v. trefwoorden) vermelden. Per referentie is het aantal abstractkaarten, evenenals het aantal titel- en locatiekaarten, beperkt tot 100.

2.3 **Identificatie**
Ter identificatie van het referentiebestand moet een tweetal identificatiekaarten worden meegegeven. Zij worden vooraan in het bestand geplaatst en vormen zodoende als het ware een label van het bestand (zie figuur 2a). De numerieke code is nul.

```
                                                                    PAGE 15

DAWSON,R.F.F.
     COST OF ROAD ACCIDENTS IN GREAT BRITAIN
     ROAD RESEARCH LABORATORY,RRL REPORT LR 79(1967)

DETROIT EDISON COMPANY.
     EMERGENCY AND GROWTH OF AN URBAN REGION: THE DEVELOPING URBAN
     DETROIT AREA VOL.1 ANALYSIS, VOL.2 FUTURE ALTERNATIVES
     DETROIT,1966-1967.

DEVINE,E.
     THE TREATMENT OF INCOMMENSURABLES IN COST-BENEFIT ANALYSES
     LAND ECONOMICS VOL.42(1966)NO.3,PP.383-386.

DICESARE,F.-SRIDHARAN,R.
     NETWORK SIMULATION CONSIDERATIONS IN THE METHODOLOGICAL DESIGN FOR
     THE PROPOSED PENNSYLVANIA DEPARTMENT OF TRANSPORTATION
     CARNEGIE-MELLON UNIVERSITY,TRANSPORTATION RESEARCH INSTITUTE(1968).

DICKEY,J.W.-STEISS,A.
     PROGRAMMING MODELS IN THE SYSTEMIC PLANNING PROCESS
     HIGH SPEED GROUND TRANSPORTATION JOURNAL VOL.3(1969)NO.1,PP.33-45.

DICKEY,J.W.
     PASSENGER TRANSPORTATION PLANNING AND DESIGN
     COUNCIL OF PLANNING LIBRARIANS,EXCHANGE BIBLIOGRAPHY NO.78(1969).

DICKEY,J.W.
     TRANSPORT SYSTEM SYNTHESIS
     COUNCIL OF PLANNING LIBRARIANS,EXCHANGE BIBLIOGRAPHY NO.80(1969).

DODSON,E.N.
     COST-EFFECTIVENESS IN URBAN TRANSPORTATION
     OPERATIONS RESEARCH VOL.17(1969)NO.3,PP.373-394.
```

Figuur 4a: Voorbeeld van een uitvoerblad van het printprogramma

2.4 Algemene tekst

In het bestand kan men voorts tekst opnemen (zie figuur 2a). De kaarten, waarop deze tekst is aangebracht en die mogen ontbreken, hebben eveneens een numerieke code gelijk aan nul. In de meeste gevallen zal deze tekst betrekking hebben op de verzameling referenties. Zij volgen in het bestand onmiddellijk na de identificatiekaarten.

3 ORGANISATIE VAN HET SYSTEEM

3.1 Algemeen

Aangezien de indruk bestaat dat menig onderzoeker gebaat zou zijn met automatisering van zijn bibliografie is bij de opzet van 'BIBLIOSYSTEM' gestreefd naar een flexibel systeem dat ook buiten het Laboratorium voor Verkeerskunde bruikbaar zou kunnen zijn.

Deze flexibiliteit is bereikt door:

a het systeem uit meerdere programma's te laten bestaan,

b de volgorde waarin de programma's gedraaid moeten worden, waar mogelijk ter beoordeling van de gebruiker van het systeem te laten,

c de controles zo min mogelijk in het referentiebestand te laten ingrijpen,

d bibliografische afspraken vrij te laten,

e als programmeertaal FORTRAN-IV te kiezen, deze taal is op praktisch iedere computerconfiguratie geïmplementeerd,

f bij het printprogramma een aantal opties, die de gebruiker al dan niet kan activeren, in te bouwen.

De programma's van 'BIBLIOSYSTEM' kunnen in 4 delen worden ondergebracht, t.w.:

Fase 1: Opbouwen en controleren van het bestand (zie fig. 1a)
Fase 2: Bijwerken van het bestand (zie fig. 1b)
Fase 3: Classificeren van het bestand (zie fig. 1c)
Fase 4: Afdrukken van het bestand (zie fig. 1d)

Daarnaast is er een hulpprogramma met behulp waarvan men kan manipuleren met referentiebestanden.

Nadat in fase 1 het bestand is opgebouwd en gecontroleerd kan men dit bestand bijwerken (muteren) indien het fouten bevat of nieuwe referenties moeten worden toegevoegd.

Het aldus verkregen bestand kan worden afgedrukt (fase 4).

Echter vaak zal men de referenties eerst willen classificeren. Dit kan geschieden in fase 3 m.b.v. het classificatie-programma. Hierna kan het geclassificeerde bestand naar wens worden afgedrukt (fase 4).

De 4 fasen zullen afzonderlijk beschouwd worden.

3.2 Fase 1: Opbouwen en controleren van het bestand

Deze fase bestaat uit 3 programma's t.w.:

BIO4: *Card to disk* van de referenties (inclusief identificatie- en informatiekaarten). De referenties worden op schijf gezet, waarbij gecontroleerd wordt of het bestandsnummer numeriek is. Er wordt een verslaglijst vervaardigd waarop de geconstateerde fouten zijn vermeld.

BIO6: *Sort*
De kaarten(records) worden gesorteerd op hun bestandsnummer.

BIO8: *Controle* van het referentiebestand.
Dit programma controleert het referentiebestand op eventueel aanwezige fouten. Alle rubrieken van de diverse records van een referentie die gecontroleerd *kunnen* worden zullen ook worden gecontroleerd. Ieder record waarin een fout is geconstateerd wordt afgedrukt op een verslaglijst, terwijl de fouten daarbij worden aangegeven.
Het controleprogramma produceert uitsluitend een verslag en wijzigt *NIET* het bestand. Indien er geen fouten zijn gevonden kan men met het bestand naar fase 3 of 4 gaan. Indien er wel fouten zijn geconstateerd gaat men met het bestand naar fase 2.

3.3 Fase 2: Bijwerken van het bestand

Met behulp van fase 2 is het mogelijk, kaarten (of te wel records) toe te voegen, te vervangen of te verwijderen.

```
                                              BLAD  28

BURTON,I.(1963A)
ACCESSIBILITY IN NORTHERN ONTARIO:AN APPLICATION OF GRAPH THEORY TO
REGIONAL HIGHWAY NETWORK
ONTARIO,1963.

BURSTALL,R.(1966A)
COMPUTER DESIGN OF ELECTRICITY SUPPLY NETWORKS BY A HEURISTIC METHOD
THE COMPUTER JOURNAL VOL.9(1966),PP.263-274.

BURSTALL,R.(1967A)
TREE SEARCHING METHODS WITH AN APPLICATION TO A NETWORK DESIGN
PROBLEM
MACHINE INTELLIGENCE VOL.1(1967),PP.65-85.

CALLAWAY,A.-CASEY,P.(1964A)
OPTIMUM STREET PATTERNS
PROCEEDINGS AUSTRALIAN ROAD RESEARCH BOARD VOL.2(1964)PART 1.

CAMPBELL,E.(1968A)
AN EVALUATION OF ALTERNATIVE LAND USE AND TRANSPORTATION SYSTEMS IN
THE CHICAGO AREA
HIGHWAY RESEARCH BOARD.HIGHWAY RESEARCH RECORD NO.238(1968),
PP.103-116.

CHAN,Y.-ET AL.(1968A)
AGGREGATION IN TRANSPORT NETWORKS:AN APPLICATION OF HIERARCHICAL
STRUCTURE
M.I.T.,DEPT. OF CIVIL ENGINEERING,RESEARCH REPORT R68-47(1968).

CHAN,Y.(1969A)
OPTIMAL TRAVEL TIME REDUCTION IN A TRANSPORT NETWORK:AN APPLICATION
 OF NETWORK AGGREGATION AND BRANCH AND BOUND TECHNIQUES
M.I.T.,DEPT. OF CIVIL ENGINEERING,RESEARCH REPORT R69-39(1969).
```

Figuur 4b: Voorbeeld van een uitvoerblad van het print-programma

Door deze 3 mogelijkheden kan men via mutaties (dat zijn bestandskaarten met in de mutatiekolom een toepasselijke mutatiecode) het bestand bijwerken.

BI10: *Card to disk* van de mutaties.
De mutaties worden op schijf gezet.

BI12: *Sorteren* van de mutaties.
De mutaties worden gesorteerd op hun bestandsnummer.

BI14: *Controle* mutaties.
De mutaties worden gecontroleerd. Er wordt o.a. gecontroleerd of de mutatiecode geldig is en of er niet meerdere mutaties met eenzelfde bestandsnummer zijn.
Niet geaccepteerde mutaties worden op een verslaglijst afgedrukt, terwijl de correcte mutaties in een uitvoerdataset worden opgenomen.

BI15: *Merge*-programma.
Dit programma heeft tot doel de geaccepteerde mutaties in het bestand aan te brengen.

Wanneer het bestand bijgewerkt is dient het bestand weer gecontroleerd te worden m.b.v. controle-programma BI08 (zie fase 1).

3.4 Fase 3: Classificeren van het bestand

Deze fase bestaat uit 1 programma t.w.:

BI26: *Classificatie* van de referenties.
Met behulp van dit programma kan men de referenties classificeren naar hun eigenschappen. Deze eigenschappen van een referentie zijn vermeld in de specificatiekaart door middel van waarden in tien attributen.
Het is eveneens mogelijk om extra identificatie- en informatiekaarten alsmede namen van klassen aan het bestand toe te voegen (zie figuur 2b).
Er bestaat een hoge flexibiliteit ten aanzien van het classificeren van de referenties. De gebruiker kan door middel van stuurkaarten aangeven welke classificatie wordt gewenst.
In de stuurkaarten worden per klasse o.a. aangegeven:

a de attributen die relevant zijn voor die klasse,
b voor elk relevant attribuut de 'gunstige' waarden voor de betrokken klasse,
c de relatie (EN- of OF-relatie) tussen de relevante attributen.

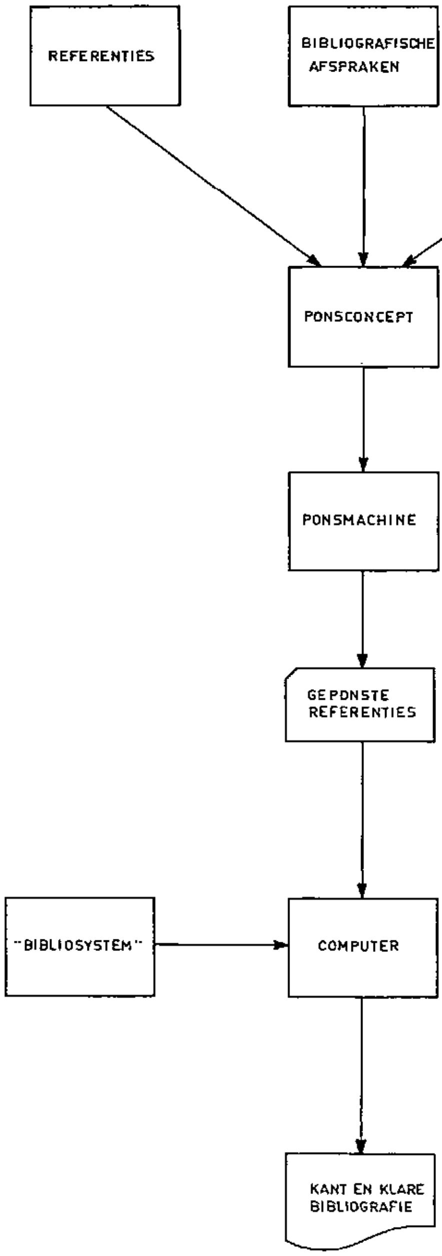
3.5 Fase 4: Afdrukken van het bestand

Deze fase bestaat eveneens uit 1 programma t.w.:

BI28: *Afdrukken* van de referenties.
Met behulp van dit programma wordt de bibliografie vervaardigd. Door middel van de hierna vermelde opties kan de lay-out vrijwel naar wens worden geregeld.
Tevens kan men regelen welke elementen van de referenties zullen worden afgedrukt (b.v. specificatiekaart en abstractkaarten). Behalve het afdrukken van de referenties en bijhouden van een bladnummering verzorgt het printprogramma ook het afdrukken van de identificatie en informatie. Bij een geclassificeerd bestand worden tevens de extra identificatie en informatie alsmede de namen van de klassen afgedrukt.
Om de bibliografie een frontpagina te geven kunnen een aantal ponskaarten die de gegevens voor deze frontpagina bevatten via het programma worden afgedrukt.

Het printprogramma bevat de volgende opties:

a De referenties afdrukken met of zonder bestandsnummer.
b De referenties afdrukken met of zonder specificatiekaart.
c De referenties afdrukken met of zonder abstractkaarten.
d Het is mogelijk achter de auteursnaam het jaar van publicatie en suffix af te drukken. Dit zal als volgt geschieden: PIETERSEN (1967A).
De toevoeging wordt dan gehaald uit de specificatiekaart. Indien de suffix blank is zal de auteursnaam als volgt worden afgedrukt: PIETERSEN (1967).
e Indien men de referenties zonder bestandsnummer wil afdrukken is er een mogelijkheid om de auteurskaart aan de linkerkant te laten uitspringen (maximaal 9 posities). Deze parameter heeft tot gevolg dat andere dan auteurskaarten het opgegeven aantal posities naar *rechts* verschuiven.
f Indien men de pagina's aan de bovenkant wil inbinden is het gewenst dat elke pagina begint met een aantal blanco regels. In de parameterkaart kan men het gewenste aantal blanco regels opgeven.
g Indien men de lijsten aan de linkerkant wil inbinden is het gewenst dat elke regel met een aantal spaties begint. Ook dit aantal kan men in de parameterkaart opgeven.
h Opgegeven moet worden welk aantal regels een pagina mag bevatten. Dit aantal mag maximaal 60 zijn.
Het is de bedoeling dat de gewenste lijsten in *A-4-formaat* geproduceerd worden. Om



Figuur 5: Van referentie tot bibliografie

het A4-formaat te localiseren wordt de linker bovenhoek van de bladen gemarkeerd door een punt. Men kan dan de geproduceerde lijsten, nadat ze al dan niet op A4-formaat zijn afgesneden, zonder meer fotocopiëren en inbinden.

i Door het printprogramma wordt hier en daar in de bibliografie enige tekst afgedrukt. In de parameterkaart kan worden aangegeven of deze tekst in de Nederlandse dan wel in de Engelse taal moet worden afgedrukt.

j Normaliter zullen de bladen genummerd worden en wel beginnend bij 1, oplopend met 1.
Wil men met een *hoger* nummer starten, dan kan dat worden opgegeven. Deze optie geeft de mogelijkheid meerdere bibliografieën samen te voegen.

3.6 **Hulpprogramma**

Het programma BI99 biedt de gebruiker van 'BIBLIOSYSTEM' een aantal mogelijkheden om te kunnen manipuleren met bestanden die door het controleprogramma BI08 accoord zijn verklaard.
Zo is het mogelijk om datasets geheel of gedeeltelijk af te drukken, te ponsen, te copiëren.
Er is voorts een optie om een lijst van auteursnamen te vervaardigen en een optie om een bestand te hercoderen.
Dit hercoderen wil zeggen dat de waarden van de attributen in de specificatiekaart, die volgens een bepaald systeem aan de referenties zijn toegekend, kunnen worden getransformeerd naar waarden die volgens een ander toewijzingssysteem worden verkregen.

HANDLEIDING

Geïnteresseerden kunnen in het bezit van de systeem- en programmabeschrijving komen door het bestellen van de publicatie 'BIBLIOSYSTEM' — Automatisering van uw bibliografieën, Laboratorium voor Verkeerskunde, Memorandum OTN/2/71.3, tegen de prijs van *f* 10,— bij: Laboratorium voor Verkeerskunde, T.H. Delft, Stevinweg 4, Delft.

SUMMARY

Introduction and relevance

On a regular basis, reports appear in the Press ahout ICT projects that fail and do not reach their original goal. This concerns both business as well as government projects. This often means significant losses with regard to investments. The failing of ICT projects is nothing new. The number of puhlications on this subject also indicates that many researchers have been or are still studying this subject. As early as 1982, Professor Jan Oonincx wrote his book "Why are information systems still failing?" [Oonincx 1982]. His conclusions were: "*Information systems, which are set up too ambitiously, too isolated or without proper planning, stand a very large chance of failing. Insufficient involvement of future users in the development of information systems or a passive attitude of the top management also often lead to disappointing results*" [Van Dijk 1982].

In his inaugural speech at the Rijksuniversiteit Groningen in 2002, Professor Egon Berghout said [Berghout 2002] the following about Jan Oonincx' publication: "This booklet is, without a doubt, based on many years of annoyance preceding 1982, bowever twenty years later it can be reprinted almost unchanged. Hardly any causes have been removed".

The American "Standish Group" has been involved for 10 years in research into ICT [The Standish Group International 2003]. In their research, they aim even more emphatically at success and failure factors regarding ICT projects. Their study, which has been appropriately baptised "*Chaos*", appears every two years. This study also shows that in 2003 only 34% are successful, 51% does not go according to plan but ultimately does lead to some result and 15% of the projects fail completely.

"As far as ICT projects are concerned, the government handles these badly". The Netherlands Court of Audit arrives at this conclusion in the research report that appeared at the end of November 2007 [Wijsman et al. 2007]. The research report of the Netherlands Court of Audit confirms that the Dutch government spends billions every single year on ICT projects that fail entirely or in part. The research was executed as a result of questions asked in the Dutch Lower Chamber. It is not clear how many projects and how much money are/is involved. The list of (partly) failed government ICT projects is a long one.

The above mentioned studies show that the subject is still very relevant in the year 2008. It may be concluded that the subject of success and failure factors in ICT projects bas been in the spotlight for more than 26 years and still is very topical.

Definition of the problem

The definition of the problem indicates the "objective in" (objective of knowledge) of the study. Based on the definition of the problem, a number of sub questions are formulated [Kuypers 1982]. In this thesis the definition of the problem is:

How were the ICT projects the author) worked on (the portfolio of projects) managed (the key here is the author's observations and experiences) with regard to success and failure factors, and how do they agree or disagree with what the procedures in Tarek Abdel-Hamid's work on Software Project Management and others say happens with regard to success and failure factors?*

*) in this chapter: the author is AvD

Sub questions

1. *What is understood by success/failure factors in ICT (for short: SUFFIs = SUccess and Failure Factors in ICT projects)?*

2. *Is it possible to derive SUFFIs from international publications, Dutch publications and from the procedures in Tarek Abdel-Hamid's work on Software Project Management and if so, what particular SUFFIs?*

3. *Are there any Big Hitters amongst the SUFFIs and if so what are these?*

4. *Which SUFFIs are applicable to what particular project from the portfolio of the author's projects?*

5. *Is it possible to use the Big Hitters for distinguishing between successful and not successful projects from the portfolio of the author's projects?*

6. *Is it possible to present the SUFFIs in a SUFFI Chart that is easy to use in practice by others?*

7. *Is it possible to apply the SUFFI Chart in the "Netherlands Court of Audit" case?*

Portfolio of projects

The portfolio consists of 9 projects with external project-based publications and 4 (project-based) audits. About the projects the author wrote 12 project-based publications in Dutch journals and 33 internal publications. The 9 projects represent an effort and duration of about 16 years. The author's role in projects 6-9 was: *internal* project manager at the Delft University of Technology. Regarding the projects 1-5, the author was the *external* project manager. Tbe 9 projects on which external publications in the trade magazines appeared and four (project-based) audits come up for discussion in chapter 5.

Definition of a project failure

The definition of a project failure is discussed in chapter 2. A successful project satisfies three factors: it complies with the functionality agreed to in advance, it is delivered on time and it is delivered within the agreed budget. When these three factors balance each other, we can speak of a successful project [Noordam et al. 2007].

For this thesis a project failure has one or more of the following characteristics:

1. it does not comply with the functionality agreed to in advance, including agreed changes of scope;
2. it exceeds the planned time-scale by more than 50%, excluding the time-scale impact of agreed changes in scope;
3. it exceeds the build cost by more than 50%, excluding the cost of agreed changes in scope.

What do others think of ICT projects failing or not failing?

In order to be able to establish how the projects the author worked on agree or disagree with what others say happens, it is necessary to map out a number of publications (what others say happens) in this field. To that purpose, the following publications will be discussed in chapter 3:

International publications

- Large Software System Failures and Successes [Jones 1996-1]
- Major Causes of Software Project Failures [May 1998]
- Critical Success Factors In Software Projects [Reel 1999]
- Seven Characteristics of Dysfunctional Software Projects [Evans et al. 2002]
- The 40 root causes of troubled IT projects [Smith 2002]
- Critical failure factors in information system projects [Yeo 2002]

Dutch publications

- Why are information systems still failing? [Ooníncx 1982]
- Success and failure factors in complex ICT projects [Beenker 2004]

- ICT project management on the road to adulthood:
  Success factors for ICT projects [Noordam et al. 2007]

Tarek Abdel-Hamid
The procedures of Tarek Abdel-Hamid and Stuart Madnick in:
"Software Projects Dynamics – An Integrated Approach" [Abdel-Hamid & Madnick 1991].

The writers of these publications report on research carried out in the field of success and failure factors of ICT projects. Some writers have also published books on this subject (for example Abdel-Hamid, Smith, Oonincx, Jones). For that reason, this collection of publications may also be considered representative for this subject as far as this thesis is concerned.

In Chapter 3, "The 40 root causes of troubled IT projects" by John Smith [Smith 2002] come up for discussion. Next, the following perspectives come up: "the size of projects" [Jones 1996-1], "project risks based" [Evans et al. 2002], "some other writers" [Yeo 2002, May 1998, Reel 1999] and "the Dutch situation" [Oonincx 1982, Beenker 2004, Noordam et al. 2007]. The procedures of Tarek Abdel-Hamid [Abdel-Hamid & Madnick 1991] are extensively discussed as well and the SUFFI model is presented (figure 4.1).

Which root causes are the "Big Hitters"?

Chapter 4 lists the views of the writers on the most important (most common) success and failure factors, followed by a conclusion. John Smith [Smith 2001] calls these factors "Big Hitters".
The Big Hitters are:

- poor project management;
- deadlines are unrealistic;
- poor communication;
- incomplete/weak definition requirements;
- insufficient involvement of future users.

In section 9.8 two other Big Hitters are introduced:

- lack of senior management involvement and commitment;
- lack of professionalism.

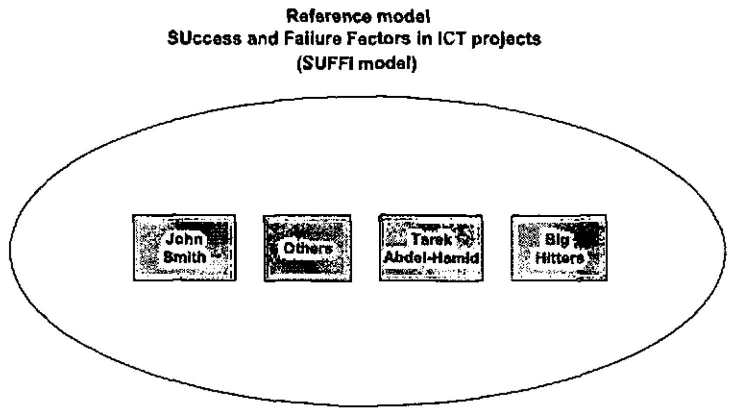


Figure 4.1: Reference model SUccess and Failure Factors in ICT projects (SUFFI model)

Which SUFFIs are applicable to what particular project from the portfolio of the author's projects?

Chapter 5 researches for each project from the portfolio which SUFFIs are applicable. For each project, a table is drawn up stating which Big Hitters are applicable, which SUFFIs related to the procedures of Tarek Abdel-Hamid are applicable and which other SUFFIs from the SUFFI model are applicable. Moreover, this chapter states what the score is with regard to the three characteristics that describe the concept of project failure, for example the Telephony project (see table 5.5.2).

Telephony	Score
Complies with functionality agreed	Yes
On time	Yes
Within the agreed budget	No *)
*) but it did not exceed the build cost by more than 50%	

Table 5.5.2: Results Telephony project

Is it possible to use the Big Hitters for distinguishing between successful and not successful projects from the portfolio of the author's projects?

Table 6.1 contains the Big Hitters in relation with the discussed cases. Although the results are based on a very limited spot check, the conclusion may be drawn that the collection Big Hitters within this collection of cases acts discriminating. Where at least four of the five Big Hitters are not applicable, the "score" is positive. Where at least four of the five Big Hitters are applicable, the "score" is negative.

This picture also corresponds with my experiences and observations during other projects and audits. When the five Big Hitters lead to a negative score, a large number of other SUFFIs usually play a part.

Results cases	Apply to					Score		
	Big Hitter 1	Big Hitter 2	Big Hitter 3	Big Hitter 4	Big Hitter 5	Funct.	On time	Within Budget
Case 1: POTVIS project (KLPD)	No	No	No	No	No	Yes	Yes	Yes
Case 2: Kolibrie project (KPN Telecom)	No	No	No	No	No	Yes	Yes	Yes
Case 3: Charging method project (GAK)	No	No	No	No	No	Yes	Yes	Yes
Case 4: Telephony project (DUT)	No	No	No	Yes	No	Yes	Yes	No
Case 5: OKAPI project (UoA)	No	No	No	No	No	Yes	Yes	---#
Case 6: GIRAF project (DUT)	No	No	No	No	No	Yes	Yes	---#
Case 7: AUBID project (DUT)	No	No	No	No	No	Yes	Yes	---#
Case 8: VDV project (DUT)	No	No	No	No	No	Yes	Yes	---#
Case 9: BIBLIOSYSTEM project (DUT)	No	No	No	No	No	Yes	Yes	---#
---	-------	-------	-------	-------	-------	-------	------	-------
Case 10: Audit Multihouse	Yes	Yes	Yes	Yes	---+	No	No	No
Case 11: Audit SYSA (GOVERN)	Yes	Yes	Yes	Yes	Yes	No	No	No
Case 12: Audit ACCINT (PUBLIC)	Yes	Yes	Yes	Yes	Yes	No	No	No
Case 13: Audit SOX (FINANCE)	No	No	Yes	No	No	Yes	Y/N*	Y/N*

+) unknown
#) no specific budget available
*) Yes or No, depends on the project

Table 6.1: Big Hitters in relation with the discussed cases

Is it possible to present the SUFFIs in a SUFFI Chart that is easy to use in practice by others?

In the previous sections, the **SUFFI model** was mentioned. It consists of tables. The SUFFI model can be put to use immediately and next further tested and improved. In order to simplify its use even more, the tables have been combined into one single table, the so-called **SUFFI Chart**. This is included in chapter 7. The SUFFI Chart was created from a number of tables. These tables include the research results of various researchers. In order to be able to test new research results more easily against all tables, a **SUFFI Total Chart** was created in which all tables were included.

Is it possible to apply the SUFFI Chart in the "Netherlands Court of Audit" case?

Case "Netherlands Court of Audit"

It has been indicated that the SUFFI model or the SUFFI-Chart as shown in chapter 7 can be used immediately and that further investigation may take place. For reasons of major social interest, the subject of success/failure factors will demand necessary attention over the next few years. Both researchers as well as project managers are invited to test the SUFFI-Chart against their own research results and practical experience. This will enable further improvement of the SUFFI-Chart, serving all those involved in ICT projects well.

At the end of November 2007, the Netherlands Court of Audit published the report "Lessen uit ICT-projecten bij de overheid – Deel A" [Wijsman et al. 2007] (Lessons from government ICT projects – Part A). Various experts have commented on this report. It is interesting to test the SUFFI-Chart against this case. Section 8.2 provides a brief summary of the report. Section 8.3 includes the following tables:

- table 8.1: main success/failure factors regarding ICT projects according to the report of the Court of Audit;
- table 8.2: recommendations/remarks from the experts as a result of the report of the Court of Audit;
- table 8.3: overview Advisors/Experts.

Table 8.1 includes the conclusions of the Court of Audit represented by 39 success/failure factors. In table 8.2, the recommendations/remarks of experts have been recorded in the shape of 65 items of recommendations/remarks. For all 104 items, it was checked which SUFFIs from the SUFFI-Chart can be related to the item in question. For each item, one or more related SUFFIs were included. In this, the main applicable SUFFIs have been shown. Completeness was not the aim. In some cases, a one on one relation applies, such as for example with regard to item AR3: "Political deadlines can be fatal to a project", which is related to SUFFI BH/02: "Deadlines are unrealistic".
In other cases, an item is related to a combination of SUFFIs. One example of this is item AR7: "Make sure to have a grip on your ICT projects", which is related to SUFFIs BH/01 ("Poor project management") and JS/RC15 ("Poor project planning, management and execution"). This meant that when the SUFFIs BH/01 and JS/RC15 are transformed from failure factors to success factors, AR7 would also become a success factor. Conclusions:

- in the tables 8.1 and 8.2 all items are related to one single SUFFI from the SUFFI Chart or to a combination of SUFFIs. On the basis of this, the SUFFI Chart does not need to be extended;
- nevertheless it is advisable to demand closer attention for some items for example:
    - **EX49/EX52**

      There is a gross lack of professionalism in the world of ICT. Only a very small section of people that are executing ICT projects at the moment have actually qualified in informatics. Surely, that is no good whatsoever! Universities should better train people in managing and executing large ICT projects [Verhoef table 8.2];

- **EX59**

  The project should be consistent with the federal and departmental information architecture [Verhoef/Clinger Cohen Act table 8.2].

Conclusion

The answer to the research question (definition of the problem):

"How were the ICT projects the author worked on (the portfolio of projects) managed (the key here is the author's observations and experiences) with regard to success and failure factors, and how do they agree or disagree with what the procedures in Tarek Abdel-Hamid's work on Software Project Management and others say happens with regard to success and failure factors?"

was given in the answers of the sub questions.

Further research

For reasons of major social interest, the subject of success/failure factors will demand the necessary attention over the next few years. Both researchers as well as project managers are invited to test the SUFFI model/chart against their own research results and practical experience. This will enable further improvement of the SUFFI model/chart, serving all those involved in ICT projects well.

I propose a thesaurus of SUFFIs (similar to the ISO 9126-standard quality model).

CURRICULUM VITAE

Aart J. van Dijk was born on 5 June 1943 in The Hague (The Netherlands). In 1959 he passed his O levels MULO-B at the Chr. MULO in Spoorwijk in The Hague. After having worked as a counter clerk at the PTT (Dutch Post) in The Hague and having accomplished his military service as a non-commissioned officer-instructor with the Marine Corps; in 1965, he joined the then Rijks Mechanische Administratie as a trainee programmer. In 1970, he became system designer at the Transportation Research Laboratory of the Delft University of Technology (DUT). Between 1975 and 1995 he worked as a project leader at the Computer Centre of the DUT and in this capacity he realised a large number of information systems amongst other things. From 1985, he combined these activities part-time with his work as an independently established IT consultant / IT auditor. From 1995, he has worked full time for his own company.

He has spent part of his spare time studying, teaching and publishing. During his career he rounded off various studies, such as Wetenschappelijk Rekenen (Scientific Mathematics) A (1968-DUT) and B (1971-DUT), Datalogy (1972-University of Twente), mathematical/informatics engineer (1976-DUT), fully qualified mathematics teacher (1978-DUT) and Executive Master of IT Auditing (1994-Erasmus University Rotterdam). He published several dozens of articles in the trade magazines and worked as an Informatics teacher in evening education for several years.

He has worked for a large number of companies and public bodies in several roles, including the role of consultant, project leader, system designer, architect, ITIL teacher, IT auditor and researcher.
He is a member of a number of professional associations, is registered IT auditor (NOREA) and certified ITIL Service Manager (EXIN). Occasionally he acts as a (visiting) lecturer.
In his work, Aart J. van Dijk is guided by the question "What value do our IT results add to the primary processes of the organisation and to the (work) satisfaction of the end users?"