A Model-Based Framework for Classifying and Diagnosing Usability Problems

Abstract

A great deal of study has been devoted to the problem of how to identify and categorize usability problems; however there is still a lack of studies dealing with the problem of how to diagnose the causes of usability problems and how to feed them back into design process. The value of classifying usability problems can be enhanced when they are interpreted in connection with design process and activities. Thus it is necessary to develop a systematic way of diagnosing usability problems in terms of design aspects and applying diagnosis results to improve design process and activities. With this issue in mind, this paper proposes a conceptual framework that supports a systematic classification and diagnosis of usability problems. This paper firstly reviews seven approaches to classifying usability problems. Then we point out the needs of adopting a model-based approach to classifying and diagnosing usability problems and of developing a comprehensive framework guiding the use of model-based approaches. We then propose a conceptual framework that specifies how a model-based classification and diagnosis of usability problems should be conducted and suggests the combined use of three different types of models, each of which address context of use, design knowledge, and design activities. Last we explain how a sound classification scheme of usability problems can be systematically developed and how the classification of usability problems can be connected to design process and activities on the basis of the framework.

Keywords: Usability Problems, Usability Problem Classification, Usability, Usability Evaluation, User Interface Design

1. Introduction

As one of the critical quality attributes of IT systems, usability has been much studied during recent decades. It is well known that systems showing a high degree of usability can ensure tangible and measurable business benefits (Peuple and Scane 2004; Strawderman and Koubek 2008). Usability can be defined as 'the capability of IT systems to be understood, learned, used and be attractive to the user, when used under specified conditions' (Schoeffel 2003). Important usability research topics include: usability factors, usability evaluation methods and metrics, user interface design principles and guidelines, usability problem classifications, and user-centred design methodologies (Te'eni et al. 2007).

Many studies have examined several usability factors characterizing the concept of usability, which makes it difficult to give an absolute definition of usability. For example, ISO/IEC 9241 (1998) specifies three dimensions: effectiveness, efficiency, and satisfaction. Nielsen (1993) gives another example of such factors: learnability, efficiency of use, memorability, errors, and satisfaction. Usability factors can be categorized into two groups: objective and subjective. Objective usability factors are concerned with the assessment of how well users perform their tasks, whereas subjective usability factors attempt to evaluate how users actually feel the usability of a system (Bevan 1999; Jin et al. 2009).

There are a lot of usability evaluation methods or techniques, and there are several ways of classifying them. However, they are usually categorized into the three groups: usability testing, usability inquiry, and usability inspection (Zhang 2011). Usability testing methods make users conduct a set of tasks by using a system or a prototype and then evaluate how they conduct their tasks. Co-discovery learning, question-asking protocol, and shadowing method are the typical examples of usability testing methods. Usability inquiry methods observe how users use a system in real work settings and ask them questions in order to understand their feelings about the system and their information needs. Field observation, focus groups, and questionnaire survey are categorized into usability inquiry methods. Usability inspection methods examine usability-related aspects in an analytic manner. Typical inspection methods contain cognitive walkthrough, pluralistic walkthrough, and heuristic evaluation. As there is absolutely no best method for all situations, it is necessary to choose an appropriate method, taking into account evaluation purposes, available time, measures to be collected, and so on. All of the three types of methods are effectively used for formative evaluation, which is conducted during the development process of a system in order to form or influence design decisions, as well as for summative evaluation, which is conducted after the development of a system is finished in order to ensure that it satisfies usability requirements or meets certain standards (Fernandez et al. 2011).

Ivory and Hearst (2001) classifies usability evaluation methods into five groups by adding two types of evaluation methods to the three types described above. The additional two types are analytical modelling and simulation that focus on the prediction of usability based on the models of users and user interfaces. Typical methods of analytical modelling include cognitive task analysis, task-environment analysis, and GOMS (Goals, Operators, Methods, and Selection Rules). Petri net models and genetic algorithm-based models are the representative examples of the models employed for simulation methods.

Although a variety of design features influence the level of usability, user interfaces would be one the most important design factors affecting usability (Freudenthal and Mook 2003). For this reason, a lot of user interface design principles and guidelines have been developed to support interface designers' development activities. Design principles are high-level design goals that hold true irrespective of task or context, whereas design guidelines are more specific rules that serve as means implementing design principles depending on task and context (Peuple and Scane 2004). Consistency is one example of design principles, and one guideline corresponding to this is 'always place home button at top left hand corner'.

Usability engineering is an organized engineering process and a set of methods that specify and measure usability quantitatively throughout the development lifecycle (Nielsen 1993). It emphasizes that usability factors should be clearly specified from the very early stage, and that usability should be a central concept of the development process. All of usability engineering activities should be coherently organized and connected to other development activities (Gillan and Bias 2001; Artman and Zällh 2005; Lin et al. 2009). A critical activity to accomplish this is to classify usability problems in a systematic way and diagnose them in connection with design process and activities (Card 1998; Hassenzahl 2000; Howarth et al. 2007; Vermeeren et al. 2008; Bekker et al. 2008). For this, a great deal of study on the classification schemes of usability problems has been conducted in the field of software engineering and human-computer interaction (HCI).

However, it seems that the previously developed classification schemes do not provide sufficient methods to associate usability problems with design process and activities (Hornbæk and Frøkjær 2005; Hornbæk 2009; Lindgaard 2006). Additionally, it is not easy to identify a theoretical basis that can encompass the core ideas of the previous classification schemes. Such a theoretical basis is needed if we want to develop another classification scheme and use existing schemes effectively (Capra and Smith-Jackson 2005). With this issue in mind, this study aims at proposing a new conceptual framework for developing a scheme for classifying and diagnosing usability problems, which can give some useful insights for improving design process and activities as well. This paper firstly reviews seven approaches to categorizing usability problems. From the review, we point out that a new conceptual framework, which offers theoretical backgrounds and practical models for developing a classification scheme, is needed for the study of usability problems. Then we describe a set of requirements for developing a new classification scheme. Next we explain the proposed framework that emphasizes the combined use of three different kinds of models, each of which addresses context of use, design knowledge, and design activities. Last we explain how a comprehensive, coherent classification scheme of usability problems can be systematically developed from the proposed framework and how the classification and diagnosis of usability problems can be bridged to design process and activities on the basis of the framework.

2. Research Backgrounds

COST action 294-MAUSE (towards the MAturation of information technology USability Evaluation) is a European consortium that was established to study usability evaluation methods and usability problem classifications in a more scientific manner (www.cost294.org) (Law 2004). MAUSE identified eight significant problems related to usability evaluation and usability engineering, which need much more research activities, based on the opinions of usability professionals working in Europe.

- A lack of a sound theoretical framework to explain the phenomena observed
- A lack of a set of empirically based and widely accepted criteria for defining usability problems
- A lack of a standard approach to estimating values of key usability test parameters
- A lack of effective strategies to manage systematically the user/evaluator effect
- A lack of a thoroughly validated defect classification system for analyzing usability problems
- A lack of widely applicable guidelines for selecting tasks for a scenario-based usability evaluation
- A lack of a sophisticated statistical model to represent the relationships between usability and other quality attributes like reliability
- A lack of a clear understanding about the role of culture in usability evaluation

These problems seem to come from the fact that many of the usability studies have been oriented to resolve practical issues without paying much attention to the development of a well-grounded scientific and methodological basis that can bridge theoretical concepts about the interaction between users and systems and the development of usable IT systems (Gillan and Bias, 2001; Hornbæk 2006; Law 2004). From the list of these problems, we can understand how important it is to classify usability problems systematically and connect them to design process and activities in order to improve the usability of IT systems.

Research on usability problem classifications and research on usability evaluation methods should be distinguished. The aim of usability evaluation methods is to offer an effective way of identifying and collecting usability problem data. However, even if usability problems are well identified and collected, there is still another important issue of classifying and interpreting usability problems. The classification schemes of usability problems have been developed to address this issue (Capra and Smith-Jackson 2005; Howarth et al. 2007). As such, it should be again noted that the purposes of usability problem classification schemes are different from those of usability evaluation methods, though they are highly interrelated.

Here we review seven approaches to classifying usability problems as follows. Although there are other classification schemes, these seven schemes seem to be sufficient to explain and compare the typical approaches for classifying usability problems (Law 2004). Of those, four schemes were originally developed to detect and classify software defects in the field of software engineering and have been adaptively used for classifying usability problems. The other three approaches, which have been developed in the field of HCI, classify usability problems based on user and interaction models.

- Orthogonal Classification Scheme (ODC) (Chillarge et al. 1992)
- Root Cause Analysis (RCA) (Leszak et al. 2002)
- Hewlett Packard Defect Classification Scheme (HP-DCS) (Huber 1999)
- Classification of Usability Problem Scheme (CUP) (Vilbergsdottir et al. 2006)
- Usability Problem Taxonomy (UPT) (Keehan et al. 1999)
- User Action Framework (UAF) (Andre et al. 2001)
- Usability Problems Classification using Cycle Interaction (CI) (Ryu and Monk, 2004)

2.1 Classification schemes based on software engineering perspective

The main purpose of categorizing usability problems is to understand the nature of the problems more systematically and thus to diagnose the probable causes of them and improve design process and activities. It is known that diagnosing the causes of a problem and applying the diagnosis results to the improvement of design process is extremely more difficult than identifying the phenomena of a defect or problem (Hollnagel 1993; Springett 1998). In the classification of usability problems, an essential activity is to determine the attributes by which usability problems can be classified and the way of how to use those attributes to characterize them. In this regard, the classification schemes of software defects in the field of software engineering can be usefully used for usability problems. For this reason, MAUSE has also studied the effective ways of applying the classification schemes of software defects to the categorization of usability problems. As MAUSE suggested, the following four classification schemes can be good references to be used for usability problem classification.

The ODC was developed by IBM in order to give system developers meaningful feedback on the progress of the current project (Chillarge et al. 1992). It is aimed to bridge the gap between statistical defect models and causal analysis; thus it strives to find out well-defined cause-effect relationships between software defects and their effects on development activities. The ODC provides a basic capability to extract diagnostic quality information from defects and infer the health of the development process from the information. It categorizes defects in accordance with the eight dimensions or factors that are the objective properties characterizing the meaning of a defect (Table 1). These factors are organized using the two process steps, in which defect classification data are collected (Freimut 2001). The process step Open is carried out when a defect was found and a new defect report is opened in the defect tracking system; however, the process step Close is performed when the defect has been corrected and the defect report is closed. As shown in Table 1, three factors are grouped to the Open process and the other five factors belong to the Close process. Each factor is again composed of several values. For example, the values of Defect Type contain assignment, checking, algorithm, function, timing, interface, and relationship. All of the factors are necessary to provide the exact semantics of a defect; however two factors Defect Type and Trigger play a significant role in the ODC.

<Table 1. Eight attributes characterizing a defect in ODC (Freimut 2001)>

The RCA is a classification scheme that was used for retrospective analysis of the defect modification requests discovered in the process of building, testing, and deploying a release of a network element as part of an optical transmission network (Leszak et al. 2002). In order to capture the semantics of a defect from the multiple points of view, the RCA has five categories: Phase Detected, Defect Type, Real Defect Location, Defect Trigger, and Barrier Analysis. The Phase Detected refers to a phase on the development life cycle including ten phases, which begin from system definition and end with deliveries. The Defect Type divides defects into three classes: implementation, interface, and external. Each class is further composed of more detailed defect types. For example, the implementation class contains the eight types of defects: data design/usage, resource allocation/usage, exceptional handling, algorithm, functionality, performance, language pitfalls, and others. The Real Defect Location specifies where a defect was located by using three values: document, hardware, and software. The Defect Trigger means the actual root causes in the RCA. The RCA assumes that there may be several underlying causes rather than just one. Four inherently non-orthogonal classes of root causes are provided: phase-related, human-related, project-related, and review-related. Like the Defect Type, these four classes are further composed of more detailed types of causes. For example, the human-related class contains the following: lack of tools knowledge, lack of process knowledge, change coordination, introduced with other repair, communication problems, missing awareness, lack of domain knowledge, lack of system knowledge, and individual mistake. The Barrier Analysis suggests measures for ensuring earlier defect detection and preventing defects.

The HP-DCS was developed to improve the development process by minimizing the number of software quality defects over time (Huber 1999). It has three descriptors for characterizing a defect: Origin (the first activity in the lifecycle where a defect could have been prevented, not where it was actually found), Type (the area, within a particular origin, which is responsible for the defect), and Mode (designator of why the defect occurred). Like the ODC and the RCA, each descriptor is composed of several factors or factor groups, of which combination classifies defects. The Origin has six factors: specification/requirements, design, code, environment/support, documentation, and others. The Type has six factor groups, and one example is a group comprising logic, computation, data handling, module interface, and so on. The Mode explains the reason of defects with five factors, which are concerned with whether information was missing, unclear, wrong, changed, or done in a better way. One important thing is that the choice of a factor for the Origin constrains the possible set of factors for the Type. In other words, the HP-DCS assumes that a factor for the Type is only meaningful in a particular phase of system development process.

The CUP was developed for the purpose of classifying usability problems to give system developers better feedback on how to correct the problems, on the basis of collective review of previous defect classification schemes described above (Vilbergsdottir et al. 2006). It specifies 10 attributes to characterize a usability problem. They include: Identifier (ID), Description, Defect Removal Activity, Trigger, Impact, Expected Phase, Failure Qualifier, Cause, Actual Phase, and Error Prevention. As in the other schemes above, most of these attributes have a set of values of their own. For example, the Cause has five values: Personal, Technical, Methodological, Managerial, and Review.

2.2 Classification schemes based on HCI perspective

The UPT is a taxonomic model in which usability problems detected in graphical user interfaces with textual components are classified from the two perspectives: artefact and task (Keehan et al. 1999). It was developed on the basis of systematic review of 400 usability problems collected in real industry projects. As shown in Fig. 1, it is made up of three hierarchical levels. The UPT has an artefact component and a task component, which are located at the top level. The artefact component focuses on the interface objects in which users experience usability problems, whereas the task component is concerned with difficulties encountered in the process of conducting a task. The two top-level components are divided into five primary categories, of which three belong to the artefact component and two pertain to the task component. Each primary category is composed of multiple subcategories. For example, visualness consists of five subcategories: object (screen) layout, object appearance, object movement, presentation of information/results, and non-message feedback (Fig. 1). The UPT categorizes usability problems into three types depending on how a problem is characterized by a combination of the categories and subcategories in Fig. 1. The three types of problems are: full classification (FC), partial classification (PC), and null classification (NC). The FC implies that a usability problem is classified in a rightmost subcategory (the deepest level in Fig. 1). The PC implies that a problem is classified in either a primary category or a subcategory that is not at the deepest level. The NC implies that no category is selected along a given component. Classification produces a pair of outcomes for the two top-level components. For example, a usability problem could be fully classified in the artefact component (FC classification) and partially classified in the task component (PC classification).

<Fig. 1. Classification of usability problems in UPT (Keehan et al. 1999)>

The UAF is an interaction model-based structure for organizing usability concepts, issues, design features, usability problems, and design guidelines (Andre et al. 2001). It thus aims to be an integrated framework for usability inspection, usability problem reporting, usability data management, and effective use of design guidelines. Another main purpose of the UAF is to support consistent understanding and reporting of the underlying causes of usability problems. Usability problem classification in the UAF employs an interaction cycle model, which adapted and extended Norman's 'stage of action' model. The cycle model is all about what users think (cognitive actions), do (physical actions), and see (perceptual actions) during the cycle of interaction with computer. It consists of four activity phases: Translation (determining how to do it with physical actions), Planning (determining what to do), Assessment (determining, via feedback, if outcome was favourable), and Physical Action (doing it). This cycle model provides a high level organization and entry points to the underlying structure for classifying usability problems. Finding the correct entry point for a usability problem is based on determining the part of the interaction cycle where the user is affected. Examples of relating usability problems to relevant part of the interaction cycle are: 'unreadable error message' and Assessment, 'user does not understand master document structure' and Planning, 'user cannot directly change a file name in an FTP program' and Translation, and 'user clicks on wrong button' and Physical Actions.

Ryu and Monk (2004) developed a classification scheme based on a cyclic interaction model, which is similar to the interaction cycle in the UAF. It is aimed at examining low-level interaction problems and thus they also developed a simple walkthrough method. The cyclic interaction model strives to model a recognition-based interaction between users and systems, by considering three paths in an interaction cycle: action-effect path, effect-goal path, and goal-action path. These three paths result in three kinds of usability problems: action-effect problems, effect-goal problems, and goal-action problems. The action-effect problems are deeply related to mode problems that lead same action to different system effects. In general, the mode problems can be classified into three groups: hidden mode problems, partially hidden mode problems, and misleading mode signals. The effect-goal problems are concerned with goal reorganization process. Ineffective or improper goal reorganization can be explained by the four types: missing cues for goal construction, misleading cues for goal construction, missing cues for goal elimination, and misleading cues for goal elimination. The goal-action problems occur when users should perform unpredictable actions to achieve a goal, which can be explained in terms of affordance. Typical two unpredictable actions are: weak affordance of a correct action and strong affordance of an incorrect action. Ryu and Monk (2004) suggested the use of walkthroughs, which is a usability inspection method, in identifying and classifying usability problems based on their proposed classification scheme.

2.3 Need of a new classification scheme and conceptual framework

Capra and Smith-Jackson (2005) consulted usability practitioners in a series of three studies and then developed a set of 10 guidelines for describing usability problems. They are as follows: (1) be clear and precise while avoiding wordiness and jargon, (2) describe the impact and severity of the problem, (3) justify the problem with data from the study, (4) describe the cause of the problem, (5) describe observed user actions, (6) describe a solution to the problem, (7) consider politics and diplomacy when writing your description, (8) describe your methodology and background, (9) help the reader sympathize with the user, (10) be professional and scientific in your description. These guidelines can be usefully used to make a yardstick for evaluating the classification schemes of usability problems.

By referring to the guidelines above, we had an interview with five usability practitioners working in a mobile phone manufacturer and a software quality certification organization. We showed those guidelines to them and asked them other requirements that they think are important in development of the classification schemes of usability problems. Collecting the guidelines developed by Capra and Smith-Jackson (2005) and the interview consultation results, we identified eight aspects for evaluating classification schemes for usability problems. Table 2 shows the evaluation aspects and the comparison between the seven classification schemes explained previously.

<Table 2. Comparison of classification schemes>

Classification schemes originated in software engineering community, which include ODC, RCA, HP-DCS, and CUP, tend to understand usability problems within a broad development context and from the perspective of developers. They have two advantages that are of benefit to the development of a classification scheme. One advantage of those schemes is to provide a range of attributes and their corresponding values, which is helpful to define a set of attributes for classifying usability problems from multiple points of view. However, this could be a disadvantage of those schemes as well. If analysts blindly follow those schemes, they are likely to miss usability problems that cannot be identified by predefined set of attributes. One way to reduce this likelihood is to make analysts understand the theoretical backgrounds underlying the predefined set of attributes. However, those schemes lack a sound theoretical background that can explain why and how those attributes are important for understanding usability problems. Moreover, the characterization of usability problems only with the attributes derived from those schemes is likely to produce rather narrow analysis results for users' actual problems, which do not characterize the problems from the perspective of users, as they do not well reflect users' task difficulties. The other advantage of those schemes is to provide some insights on the way of linking the classification and diagnosis of usability problems to the improvement of design process and activities. More specifically, the insights we obtained include: (1) a usability problem can be traced to its related design processes and activities through causal chains, (2) the relationship between usability problems and design activities can be many-to-many, and (3) characterization of a usability problem by using a set of attributes is needed to judge whether it should be diagnosed in connection with design aspects or not, and so on.

Classification schemes developed in the field of HCI, such as UPT and UAF, identify and categorize usability problems on the basis of interaction model or users' task model. Accordingly, in comparison with the schemes coming from software engineering community, they are more useful for understanding why users feel difficulties in usability problems and what kinds of difficulties happen from the cognitive perspective of users. However, they do not provide sufficient methods for connecting usability problems to design process and activities. In addition, system designers and evaluators need to be well aware of HCI models, such as the cyclic interaction model, in order to make use of the classification schemes coming from the field of HCI. Thus it could be a burdensome work for system designers and evaluators to use those schemes if they do not have sufficient knowledge on the models.

In order to make a usability classification scheme more practical and effectual for improving design process and activities, it should take two different perspectives (the user and the developer perspective) synergistically when characterizing usability problems. Additionally, in order to bridging between the classification and diagnosis of usability problems and the improvement of design process and activities, a scheme should be established based on a conceptual framework that weaves usability problems systematically into design process. However, it is hard to say that those two points are well addressed in the classification schemes that have been developed so far. Thus it is necessary to develop a new classification scheme, taking account of those two points and the eight evaluation aspects listed in Table 2. Here we need to consider a right way of developing a classification scheme. As shown in Fig. 2, a conceptual framework needs to be firstly developed, from which classification schemes can be systematically developed. Blandford et al. (2004) made a distinction between a particular model and the underlying framework in the development of programmable user model. They stated that the framework encapsulates a generalized collection of guidelines for construction a model. Thus any model developed from that framework inherits the same concepts and principles. Simply instantiating a framework results in a particular model. This study made an analogous distinction between a classification scheme and the conceptual framework. The conceptual framework offers the concepts, perspectives, and their relevant models that will ensure the consistent development of classification schemes over a range of IT-based systems. As a classification scheme has different features depending on the frameworks or models used for developing it, a new scheme and its usage method need to be developed in the manner shown in Fig. 2. For the reasons above, the establishment of a new conceptual framework is very meaningful to the study of usability problem classification schemes. With this issue in mind, this study is aimed to propose a new conceptual framework that emphasizes a combined use of three types of models, each of which addresses context of use, design knowledge, and design activities.

<Fig. 2. Research method for developing a new classification scheme and conceptual framework>

3. Development of a conceptual framework

3.1 Requirements for developing a new classification scheme

We derived a set of requirements for developing a new classification scheme on the basis of the comparative review of existing classification schemes, the eight aspects for evaluating classification schemes in Table 2, and the opinions of usability practitioners in the industry. These requirements serve as a conceptual basis for the framework proposed in this study. First, the concept and scope of usability should be clearly defined in accordance with the purpose of usability problems classification. Traditionally, usability concept does not include aspects related to the usefulness of system functions. However, the broader concept of usability includes usefulness (Te'eni et al. 2007). If the purpose of classification is to improve design process and activities, the broader concept of usability including the concept of usefulness should be adopted. However, if the purpose is only concerned with usability testing of certain user interface features and interface-based operations, the narrow traditional usability concept would be sufficient.

Second, a usability problem needs to be characterized in terms of basic 5W1H questions as follows. These questions make it possible to analyze usability problems from multiple points of view. The classification and diagnosis of usability problems can be regarded as attempts to seek plausible answers to these questions.

- What are the functions affected by or resulting in the usability problem?
- Who experienced the usability problem?
- When did users experience the usability problem?
- How did users undergo the usability problem?
- Which part of user interfaces (Where) is related to the usability problem?
- Why did the usability problem occur?

Third, two types of usability problems need to be distinguished. The first type is usability problems that can be objectively measured without regard to the subjective nature of a user. The second type is usability problems that are difficult to be objectively measured because they are highly dependent on the subjective feelings of a user. This issue is related to the difference between usability and user experience. Although some people still use both the terminologies interchangeably, user experience is broader concept than usability (Saffer 2007). Usability is mainly concerned with the designed features of interactive systems in terms of how easy it is to use and how useful it is. However, user experience is concerned with the user's entire interaction with products, as well as the thoughts, feelings, and perceptions that result from the interaction (Tullis and Albert 2008). Referring to the terminology from the field of software quality, we can say that usability is more related to external quality and user experience is more to the concept of quality-in-use (Bevan 1999). In this regard, the second type of usability problems can be said to be user experience problems, rather than usability problems. What is important is that we cannot design a user experience; we can only design for a user experience (Rogers et al. 2011). However, usability can be engineered as the terminology 'usability engineering' implies. Thus the first type of usability problems can be diagnosed and improved based on the design knowledge of a system. For example, the too small size of a label would be the first type of usability problems in that it is problematic to all users irrespective of an individual's perception and can be objectively measured. The design rationale underlying the too small size can be found in the design knowledge of user interface objects. However, a color of a label would be problematic only to a certain ethnic group for whom the use of the color is prohibitive for a religious reason. Also, it would be problematic only to a certain user group who don't like the color. Therefore, different approaches are needed to resolve these two different types of usability problems. However, this study is mainly concerned with the first type of usability problems.

Fourth, related to the second requirement, it should be noted that usability problems can be categorized by several criteria. From the six questions, we can identify the six corresponding criteria as follows: system function criteria (what), user criteria (who), task criteria (when), interaction step criteria (how), interface object/feature criteria (where), and design principle criteria (partly why). For example, if usability problems of a word processor are classified in terms of task, some can be related to editing tasks while others can be concerned with formatting tasks. If they are categorized by interface object criteria, some can be regarded as menu-related problems and others can be information architecture-related problems. As such, a usability problem can be flexibly classified with the six criteria.

Fifth, if we want to use usability problems more effectively to improve design process and activities, usability problems and design activities should be able to be described by a same language or modeling concept. As will be explained later, the proposed framework recommends the consideration of the abstraction level of design knowledge to satisfy this requirement.

3.2 New conceptual framework

Taking into account the requirements above, we propose a new conceptual framework from which new classification schemes can be systematically developed and in which existing classification schemes can be compared more methodically (Fig. 3). Several conceptual frameworks that can satisfy the five requirements described in section 3.1 could be developed. However, this study proposes the framework being composed of the three perspectives: Context of Use, Design Knowledge, and Design Activities; thus it stipulates that a usability problem needs to be analyzed in the collective consideration of the three perspectives. Relating the three perspectives to the five requirements, we can say that (1) the first, second, third requirements could be addressed by the Context of Use perspective, (2) the fourth requirement could be partially addressed by the Context of Use perspective and partially handled by the Design Knowledge perspective, and (3) the fifth requirement could be dealt with by the integrated consideration of the Design Knowledge and the Design Activities Perspectives.

The analysis of usability problems from each perspective can be facilitated by the use of models that reflect the perspective. As there could be several models related to each perspective, there is no absolute answer to the question of what model should be used. However, as shown in Fig. 3, the proposed framework recommends the use of the three models: Artefact-Users-Tasks-Organization-Situation (AUTOS) model for Context of Use, Abstraction Hierarchy (AH) model for Design Knowledge, and Function-Behaviour-Structure (FBS) model for Design Activities.

<Fig. 3. Proposed conceptual framework>

This framework emphasizes that usability problems should be interpreted with a proper understanding of context of use, with particular attention paid to users' cognitive tasks and knowledge use. Users' understanding on the design knowledge of a system, which they interact with, is an important cue for examining how users conduct their tasks and why and how they experience usability problems. Their understanding on the design knowledge may be the same as or different from the system designer's design knowledge. The comparison between them can be a useful basis for relating usability problems to design process and activities. Such a relationship cannot be absolute and completely explained. However, it is meaningful to identify such a relationship in that it gives system designers and evaluators a conceptual basis for diagnosing usability problems in association with design process and activities. Most of the existing classification schemes focus only on determining criteria and classifying usability problems based on the criteria. Thus they do not offer a useful way of making use of usability problems during design process. In this regard, the proposed framework can be said to be an attempt to overcome the limitation of existing classification schemes.

3.2.1 Context of use perspective

When a usability problem occurs, it cannot be correctly classified and diagnosed without properly understanding the context of use in which it happened (Lavery et al. 1997). For this reason, it can be said that systematic analysis of contexts of use is the most fundamental process for characterizing usability problems (Savioja and Norros 2012). As a context of use is regarded as a kind of emergent property, it is impossible to analytically identify all factors constituting a context of use (Heo et al. 2009). Additionally, even if they are all identified, a context of use cannot readily be described simply by examining them in isolation or even some of them together. Nevertheless, many researchers agree that identifying the factors underlying contexts of use and their interrelationships is still the basis of the most systematic ways of fathoming the nature of usability problems (Howarth et al. 2007; Law 2004; Vermeeren et al. 2008).

In general, some assumptions about the interaction between users and systems are needed to analyze contexts of use. Factors underlying contexts of use can be differently identified depending on what assumptions are made. The way that users interact with systems is deeply related to the process of their formulating task goals. Regarding this, we can think of two different assumptions. One assumption is that user task goals are formulated reactively in accordance with the responses or feedbacks of systems against user actions. This assumption is well reflected in the classification schemes developed in the field of HCI, such as the UAF (Andre et al. 2001) and the CI (Ryu and Monk 2004). However, some of the recent studies in HCI pointed out the limitations of this assumption and showed that user goal formulation is influenced by not only the responses of systems but also other factors including users' internal cognitive processing and environmental conditions (Boy 1998). This second assumption, which users formulate their task goals proactively rather than reactively, seems to be more valid for the analysis of contexts of use.

For this reason, the proposed framework employs the AUTOS model that was developed by Boy (1998). As shown in Fig. 4, this model attempts to explain contexts of use with the combination of five factors: artefact, user, task, organization, and (physical) situation. Additionally, he developed a cognitive task analysis method called cognitive function analysis (CFA) that helps analyze the cognitive interaction between users and systems in a more detailed way. We mentioned previously that the six criteria could be effectively used for the classification of usability problems. The analysis of contexts of use based on the AUTOS model makes it possible to consider four criteria (task, user, interface object, and interaction step) simultaneously and thus offers a useful means for interpreting usability problems.

<Fig. 4. AUTOS model (adapted from (Boy 1998))>

When we interpret usability problems by using the AUTOS model, we can think of several possible cases as follows. A usability problem can be critical to all users, whereas another problem can be meaningful to a particular user (group). A usability problem can have nothing to do with the features of systems but is only related to organizational or situational factors. A usability problem can happen only when conducting a particular task. A usability problem can happen only when a particular user (group) conduct a particular task. In this way, usability problems can be variously but systematically interpreted by the use of the AUTOS model and the four criteria. However, when a usability problem is more related to the task and the interface object criteria, it can be more readily connected to the Design Knowledge perspective. Thus two different perspectives (the user and the developer perspective) can be taken together more meaningfully for the usability problem.

3.2.2 Design knowledge perspective

As described above, the four criteria among the six criteria for characterizing usability problems are addressed in the Context of Use perspective. The other two criteria (system function and design principle) can be considered from the Design Knowledge perspective. Design knowledge in this framework means the knowledge on the function, behaviour, and structure of a system that users interact with. The knowledge on the function mainly represents what kinds of functions are available and how they are interrelated each other. The knowledge on the structure mainly expresses the spatial layout of user interface components or objects and their appearance. The knowledge on the behaviour mainly describes how user interface objects work in order to accomplish the designed functions. Taking account of the design knowledge of a system, it is possible to relate these two criteria to the task criteria and the interface object criteria. The usability problems related to task and interface designs are likely to be concerned with the errors in the design of system functions, interface features, and the application of design principles, all of which can be inferred from the design knowledge of a system. There are several models available to represent the design knowledge of a system. However, the proposed framework recommends the AH model that was developed by Rasmussen (1985).

The AH model is a multilevel knowledge representation model for describing the functional structure of a work domain or system. It was originally developed to represent the design knowledge of a complex process control system in the field of cognitive systems engineering (CSE). However, it has been effectively used for representing several other types of systems, which include library systems, information appliances, mobile phones, automobiles, and software systems, in the field of CSE and HCI (Hassenzahl 2000; Kwon et al. 2007; Vicente 2002).

One particular feature of the AH model is that it is defined by many-to-many goal-means relationships between adjacent levels. In other words, when there is a particular function at one level, its higher level explains the reasons why the function is designed, whereas its lower level illustrates how the function is actually implemented. In the abstraction levels of the AH model, higher levels contain the information about the functional purpose of a system and lower levels describe the information about the physical implementations of its functions.

The concept of AH is conceptually differentiated from the concept of part-whole physical decomposition that is commonly used in engineering systems. Fig. 5 shows a matrix to represent the knowledge of a system, which is composed of two dimensions: functional abstraction levels and part-whole decomposition levels. This matrix makes it easier to represent how top-level functional purposes of a system are implemented into visible physical features or interface objects.

<Fig. 5. Abstraction hierarchy of a system (adapted from (Rasmussen 1985))>

There is no absolute answer to the number of abstraction levels of the AH model. However, several studies showed that five abstraction levels are effective and meaningful for modelling most of the technical systems that users need to interact with (Kwon et al. 2007). From the top level, the five abstraction levels are: functional purpose (FP), abstract function (AF), general function (GF), physical function (PF), and physical form (P). Table 3 explains the general meaning of each abstraction level and its meaning adapted for software systems.

<Table 3. Meaning of five abstraction levels>

A simple example would better clarify how a usability problem can be analyzed from the AH perspective. The 'file processing' of a word processing software is considered a function at the GF level. Several functions at the PF level are needed to realize the purpose of the 'file processing', which include 'opening file', 'closing file', 'printing file', 'saving file', etc. If a usability problem is observed at the functions of 'opening file' and 'saving file', we can hypothesize that 'file processing' function is also problematic, and that there would be likely to be a design error in the process of implementing the goal-means relations between 'file processing' function and the two functions. According to the studies on the AH, users' task situations in the interaction with a system can be effectively identified from the set of the GF-level functions (Naikar et al. 2006). Generally, a function at the GF-level can be regarded as a task that users conduct to achieve the goal of an activity by interacting with a system. For this reason, it can be said that utilizing the AH enables evaluators to put usability problems in the wider context of the whole activity that a system is supposed to support.

3.2.3 Design activities perspective

The Design Activities perspective, connected to the Design Knowledge perspective, enables us to hypothesize some ways of improving design process and activities from the classification and diagnosis of usability problems. Design theories have a viewpoint that a design process comprises a series of activities transforming the functions of a system through its behaviours to its structures (Pahl et al. 2007). Related to the concept of the AH, this viewpoint can be interpreted as that design process and activities are the attempts to transform the top-level functional purposes of a system through the three middle functional levels to the bottom-level physical forms. However, such transformations are not made in a unidirectional way. Thus they can be made from the lower-level to the higher-level, as well as from the higher-level to the lower-level. The FBS model, which was developed by Gero, systematically explains the nature of design process and activities in relation to those transformations (Gero and Kannengiesser 2004). It is a very comprehensive, generalizable model for describing design process and activities. Thus the proposed framework recommends the use of the FBS model to reflect the Design Activities perspective.

The FBS model distinguishes the three design aspects of a system or artefact: function, behaviour, and structure. In the FBS model, the function of a system is defined as its teleology (what the system is designed for). The behavioural aspect of a system describes the attributes that are derived or expected to be derived from the structures of the system. This means that the behaviours of a system are related to what a system actually does. The structural aspect of a system describes the visible components of a system and their relationships. Thus it is concerned with the actual appearance of a system. In this regard, it is very similar to the concept of the AH model; this similarity gives some insights on how to bridge the Design Knowledge perspective and the Design Activities perspective, which can consequently provide a more scientific method of connecting usability problems with design process and activities.

As shown in Fig. 6, the FBS model defines the elementary five states in the design process, which are function (F), expected behaviour (Be), behaviour derived from structure (Bs), structure (S), and documentation (D). It also defines the eight abstract design activities (transformations) that link the five states (e.g., formulation). Each of the five states does not completely correspond to each abstraction level of the AH model. However, considering the meaning of each abstraction level of the AH model, it can be said that F and Be correspond to the three high abstraction levels (F, AF, and GF), Bs corresponds to PF, and S corresponds to P. By using such relationships between the two models, if a usability problem is observed, we can diagnose from the perspective of design knowledge of a system and then reason what design process and activities are more related to the problem. The relationships between usability problems and design process and activities that are obtained through such a reasoning process cannot be absolute and correct; they are only hypothetical. However, it is inevitable to conduct such an analysis and reasoning in order to improve design process and activities on the basis of usability problems. Unfortunately, existing usability classification schemes do not provide a useful way of doing this work. In this regard, we can find the usefulness of the proposed framework.

<Fig. 6. FBS model (adapted from (Gero and Kannengiesser 2004))>

4. Use of the conceptual framework

It should be again noted that the proposed framework is not a classification scheme of usability problems. Thus it is not meaningful to compare it with the existing classification schemes that were reviewed previously. The main purpose of the framework is to give a conceptual background for developing a new classification scheme. Considering the difference between a framework and a classification scheme, we can say that the framework proposed in this study is not specific to a particular context but aimed to be general. However, classification schemes derived from the framework may be specific to a particular context by taking more into consideration the peculiar contextual information of a certain context when considering the Context of Use perspective. For example, when we develop a classification scheme for usability problems in a safety-critical domain, we need to pay particular attention to the domain-specific information and features, which are different from those of other domains. Although simple examples introduced in this paper are related to general-purpose software, it needs to be addressed that classification schemes specific to a particular context may be developed from the framework.

Based on the proposed framework and the set of requirements for establishing a classification scheme, we can identify a minimal set of attributes that a new classification scheme should include (Table 4), which needs to be included in any classification schemes. However, this set of attributes is never a complete one; one can develop another classification scheme based on the framework, which contains a more comprehensive, coherent set of attributes.

<Table 4. Minimal set of attributes to be included in a classification scheme>

Except for the general two attributes, all the other attributes can be grouped into three categories, each of which addresses one of the three perspectives in the framework. All of the attributes are also categorized into either required one or optional one. The required attributes should absolutely be used to characterize usability problems from the Context of Use perspective. The resultant characteristics of a usability problem reported determine what optional attributes should be accordingly used. From the resultant characteristics, we can make a decision whether it is worth considering a usability problem in connection with the errors of design process and activities. If a usability problem is considered more related to other factors rather than the errors of design process and activities, it is unreasonable to diagnose the problem from the perspective of design process and activities.

For example, let's suppose that many people in an area complain about the use of some functions of a mobile phone that need a stable Internet connection. In this case, the main cause of this usability problem would be related more to the unstable Internet connection condition in this area than to the design deficiencies of those functions. Thus it is not necessary to connect this problem with design process and activities. We can find another example in the use of decision support software in a company. Let's suppose that the software is developed to support the decision makers whose works are based on a business process standardized in the company. However, if the inventory management workflow of the business process is changed as a result of business engineering, employees who should make a decision on the inventory management would complain usability problems in using the software. In this case, it is not meaningful to link the usability problems with the process and activities of designing the software. On the contrary, if some people report that they have difficulty in managing emails with a mobile phone, it would be likely that the main cause of the usability problem reported is related to some errors occurring in the process of designing the functions for managing emails. Thus it is worth associating the cause of the usability problem with the design process and activities of the mobile phone.

If the characterization of usability problems using the Context of Use perspective indicates that a problem needs to be further analyzed from the design perspective, we can diagnose it from the Design Knowledge perspective by using the three attributes, as shown in Table 4. Particularly, it is important to hypothesize what types of transformations in the AH model are highly related to a usability problem. Through this diagnosis process, the causes of a usability problem are reasonably linked to design process and activities. For example, if a usability problem is about the small font size, it could be related to the transformation from the PF level to the P level in the AH model.

After diagnosing a usability problem from the Design Knowledge perspective, the next step is to identify design activities that would be likely to be related to the problem from the Design Activities perspective. For this, the corresponding relationships between the FBS model and usability-related design activities, shown in Table 5, would be effectively used. Considering the purpose of usability-related design activities and referring to the relevant literatures (Kruchten 2005; Nielsen 1993; Shneiderman and Plaisant 1998; Te'eni et al. 2007), we established such relationships. However, these relationships are never complete and would not be so exclusively one-to-one. Thus they should be used only as a referential basis for connecting a usability problem with design activities. For example, in the earlier example of small font size, if the problem is considered being related to the transformation from the PF to the P, this transformation corresponds to the synthesis activity in the FBS model. Additionally, based on the relationships shown in Table 5, we can think of that the problem is highly likely to be related to the design errors of presentation design activity.

<Table 5. Relationships between FBS model-based design process and usability-related design activities >

5. Concluding remarks

This study proposed a conceptual framework that aims to support the development of a classification scheme of usability problems. As the usability of IT systems are increasingly important, the classification of usability problems and the diagnosis of them in connection with design process and activities have accordingly become significant. In this paper, we reviewed seven methods for classifying usability problems, which well reflect the typical approaches developed in the fields of software engineering and HCI. The review results pointed out that most of the existing classification schemes do not provide an effective way of connecting the classification of usability problems with the improvement of design process and activities. This study claimed that a conceptual framework should firstly be developed to address this issue, which offers theoretical backgrounds and practical models for developing a classification scheme. To develop a conceptual framework more systematically, we identified a set of requirements for establishing a new classification scheme of usability problems. Taking account of these requirements and the limitations of existing classification schemes, we developed a new framework in which a new classification scheme can be systematically developed and the earlier studies on usability problem classification can be compared more methodically.

The proposed framework emphasizes the integrated consideration of the three perspectives: Context of Use, Design Knowledge, and Design Activities. It specifies that a usability problem needs to be categorized and diagnosed in a collective consideration of the three perspectives. The analysis of usability problem from each perspective can be facilitated by the use of models that deal with the perspective. Although several models could be used for each perspective, the proposed framework recommends the use of the AUTOS model, the AH model, and the FBS model for the Context of Use, the Design Knowledge, and the Design Activities perspective respectively. The Context of Use perspective helps us to interpret usability problems within a broad context of use. It specifies that a usability problem needs to be identified as one of two types, depending on whether the causes of the problem are considered more related to design factors (e.g. inconsistent layout of user interface objects) or non-design factors (e.g. an individual's preference for a color). If a usability problem is judged more related to design factors, it should be further analyzed from the Design Knowledge perspective. This perspective helps us to diagnose the causes of the problem in terms of the design principles and the abstraction levels of the design knowledge of a system that users interact. Based on the diagnosis results, the Design

Activities perspective enables us to reason what types of usability-related design activities are likely to be related to the causes of a usability problem.

Considering the limitations of existing classification scheme mentioned above, we can say that the proposed framework is meaningful in that it attempts to connect the classification of usability problems with the improvement of design process and activities and provides conceptual foundations useful for the systematic diagnosis of usability problems. However, it should be noted that the framework is not a kind of a classification scheme. The purpose of the framework is to be used as a conceptual basis for developing a new classification scheme and dealing with usability problems; therefore, it should be regarded as a thinking tool for system designers and evaluators. Although this paper presented a set of attributes that a classification scheme should include, one can think of other attributes and then develop a more comprehensive, coherent classification scheme. Therefore, the practical process of establishing a classification scheme from the framework remains as a future research. Additionally, more case studies of using the framework in various systems should be secured in order to refine and improve it. Particularly, we expect that a more effective way of integrating the three perspectives, which offers a more practical method of improving design process and activities based on usability problems, can be obtained from more case studies.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (Grant Code: 2012042146).

References

Andre T, Hartson H, Belz S, McCreary F (2001) The user action framework: a reliable foundation for usability engineering support tools. International Journal of Human-Computer Studies 54(1):107-136

Artman H, Zällh S (2005) Finding a way to usability: procurement of a taxi dispatch system. Cognition, Technology & Work 7(3):141-155

Bekker MM, Baauw E, Barendregt W (2008) A comparison of two analytical evaluation methods for educational computer games for young children. Cognition, Technology & Work 10(2):129-140

Bevan N (1999) Quality in use: meeting user needs in quality. The Journal of Systems and Software 49(1):89-96

Blandford A, Butterworth R, Curzon PP (2004) Models of interactive systems: a case study on programmable user modelling. International Journal of Human-Computer Studies 60(2):149-200

Boy G (1998) Cognitive function analysis. Ablex Publishing Corporation, Stamford.

Capra MG, Smith-Jackson TL (2005) Developing guidelines for describing usability problems. Technical Report #ACE/HCI-2005-002, Virginia Tech

Card D (1998) Learning from our mistakes with defect causal analysis. IEEE Software 15(1):56-63

Chillarge R, Bhandari I, Chaar J, Halliday M, Moebus D, Ray B, Wong M (1992) Orthogonal defect classification-a concept for in-process measurements. IEEE Transactions on Software Engineering 18(11):943-956

Gero JS, Kannengiesser U (2004) The situated function-behaviour-structure framework. Design Studies 25(4):373-391.

Gillan D, Bias R (2001) Usability science I: Foundations. International Journal of Human-Computer Interaction 13(4):351-372

Fernandez A, Insfran E, Abrahäo S (2011) Usability evaluation methods for the web: a systematic mapping study. Information and Software Technology 53(8):789-817

Freimut B (2001) Developing and using defect classification schemes. IESE-Report No. 072.01/E, Fraunhofer IESE

Freudenthal A, Mook HJ (2003) The evaluation of an innovative intelligent thermostat interface: universal usability and age differences. Cognition, Technology & Work 5(1):55-66

Hassenzahl M (2000) Prioritizing usability problems: data-driven and judgement-driven severity estimates. Behaviour & Information Technology 19(1):29-42

Heo J, Ham D-H, Park S, Song C, Yoon, W. C. (2009) A framework for evaluating the usability of mobile phones based on multi-level, hierarchical model of usability factors. Interacting with Computers 21(4):263-275

Hollnagel E (1993) The phenotype of erroneous action. International Journal of Man-Machine Studies 39(2):1-32

Hollnagel E (1998) Cognitive reliability and error analysis method. Elsevier, Oxford

Hornbæk K, Frøkjær E (2005) Comparing usability problems and redesign proposals as input to practical systems development. In: Proceedings of ACM conference on human factors in computing systems, pp. 391-400

Hornbæk K (2006) Current practice in measuring usability: challenges to usability studies and research. International Journal of Human-Computer Studies 64(2):79-102

Hornbæk K (2009) Dogmas in the assessment of usability evaluation methods. Behaviour & Information Technology 29(1):97-111

Howarth J, Andre T, Hartson R (2007) A structured process for transforming usability data into usability information. Journal of Usability Studies 3(1):7-23

Huber J (1999) A comparison of IBM's orthogonal defect classification to Hewlett Packard's defect origins, types, and modes, Hewlett Packard Company

ISO/IEC (1998) Ergonomic requirements for office work with visual display terminal-part 11: Guidance on usability, ISO/IEC 9241

Ivory M, Hearst M (2001) The state of the art in automating usability evaluation of user interfaces. ACM Computing Survey 33(4):470-516

Jin, BS., Ji YG, Choi K, Cho G (2009) Development of a usability evaluation framework with quality function deployment: from customer sensibility to product design. Human Factors and Ergonomics in Manufacturing & Service Industries 19(2):177-194

Keehan S, Hartson H, Kafura D, Schulman R (1999) The usability problem taxonomy: a framework for classification and analysis. Empirical Software Engineering 4(1):71-104

Kruchten P (2005) Casting software design in the function-behaviour-structure framework. IEEE Software 22(2):52-58

Kwon S, Ham D-H, Yoon, WC (2007) Evaluation of software usability using scenarios organized by abstraction structure. In: Proceedings of ECCE 2007, pp. 19-22

Lavery D, Cockton G, Atkinson M (1997) Comparison of evaluation methods using structured usability problem reports. Behaviour & Information Technology 16(4-5):246-266

Law E (2004) Proposal for a new COST action 294: towards the maturation of IT usability evaluation, COST Office

Leszak M, Perry D, Stoll D (2002) Classification and evaluation of defects in a project retrospective. The Journal of Systems and Software 61(3):173-187

Lin C-H, Hwang S-L, Wang EM-Y, Pen S-L (2009) Design for usability on supply chain management systems implementation. Human Factors and Ergonomics in Manufacturing & Service Industries 19(5):378-403

Lindgaard G (2006) Notions of thoroughness, efficiency, and validity: are they valid in HCI practice? International Journal of Industrial Ergonomics 36(12):1069–1074

Naikar N, Moylan A, Pearce B (2006) Analysing activity in complex systems with cognitive work analysis : concepts, guidelines and case study for control task analysis. Theoretical Issues in Ergonomics Science 7(4):371-394

Nielsen J (1993) Usability engineering. AP Professional, San Diego

Pahl G, Beitz W, Feldhusen J, Grote K-H (2007) Engineering design: a systematic approach (3rd ed.). Springer, London

Peuple JL, Scane R (2004) User interface design. Crucial, Exeter

Rogers Y, Sharp H, Preece J (2011) Interaction design: beyond human-computer interaction (3rd ed.). John Wiley & Sons, Chichester

Rasmussen J (1985) The role of hierarchical knowledge representation in decision making and system management. IEEE Transactions on Systems, Man, and Cybernetics 15(2):234-243

Ryu H, Monk A (2004) Analysing interaction problems with cyclic interaction theory: low-level interaction walkthrough. Psychology Journal 2(3):304-330

Saffer D (2007) Designing for interaction: Creating smart applications and clever devices. New Riders, Berkeley

Savioja P, Norros L (2012) Systems usability framework for evaluating tools in safety-critical work. Cognition, Technology & Work (To be published)

Schoeffel R (2003) The concept of product usability: a standard to help manufacturers to help consumers. ISO Bulletin March:5-7

Shneiderman B, Plaisant C (1998) Designing the user interface: Strategies for effective human-computer interaction (5th ed.). Pearson Higher Education, Boston

Springett M (1998) Linking surface error characteristics to root problems in user-based evaluation studies. In: Proceedings of AVI 1998, pp. 102-113

Strawderman L, Koubek R (2008) Human factors and usability in service quality measurement. Human Factors and Ergonomics in Manufacturing & Service Industries 18(4):454-463

Te'eni D, Carey J, Zhang P (2007) Human-computer interaction: developing effective organizational information systems. John Wiley, Hoboken

Tullis, T, Albert, B (2008) Measuring the user experience. Morgan Kaufmann, Burlington

Vermeeren APOS, Attema J, Akar E, Ridder H, von Doorn AJ, Erbu C, Berkman AE, Maquire MC (2008) Usability problem reports for comparative studies: consistency and inspectability. Human-Computer Interaction 23(4):329-380

Vicente KJ (2002) Ecological Interface Design: progress and challenge. Human Factors 44(1):62-78

Vilbergsdottir S, Hvannberg E, Law E (2006) Classification of usability problems (CUP) scheme: augmentation and exploitation. In: Proceedings of NordiCHI 2006, pp. 281-290

Zhang Z (2011) Overview of usability evaluation methods. http://www.usabilityhome.com. Accessed 10 October 2011

List of Tables

List of Figures

Table 1. Eight attributes characterizing a defect in ODC (Freimut 2001)

Attributes	**Description**	**Process**
Activity	When did you detect the defect?	Open
Trigger	How did you detect the defect	
Impact	What would have customer noticed if defect had escaped into the field?	
Target	What high level entity was fixed?	Close
Source	Who developed the target?	
Age	What is the history of the target?	
Defect Type	What had to be fixed?	
Defect Qualifier	Is the defect type missing, incorrect, or extraneous?	

Table 2. Comparison of classification schemes

Evaluation Aspect	ODC	RCA	HP-DCS	CUP	UPT	UAF	CI
Theoretical backgrounds on usability	-	-	-	-	++	++	++
Multiple attributes for characterizing problems	++	++	++	++	-	-	-
Characterizing actual problems from the perspective of users	-	-	-	+	+	++	++
Identifying and analyzing the context of use in which problems are found	-	-	-	-	+	+	+
Means for linking problems to design process	-	-	-	-	-	-	-
Diagnosing the likely causes of problems from the perspective of design process and activities	+	+	+	+	+	-	-
Evaluating the impacts of problems	+	-	-	+	+	-	+
Providing solutions to problems	-	-	-	+	-	-	-

Note) -: Weak, +: Moderate, ++: Good

Table 3. Meaning of five abstraction levels

Abstraction Level	General Meaning	Meaning Adapted for Software Systems
FP	The purpose for which a system is designed	The purpose for which a software is designed
AF	The causal structure of the process in terms of mass, energy, information, or value flows; the values or priorities that must be preserved in carrying out the work of a system	The values or priorities that must be preserved in carrying out tasks supported by a software
GF	The purpose-related functions that a system is designed to achieve	The purpose-related functions that a software is designed to achieve the functional purpose
PF	The characteristics of the components and their interconnections	The user interface object-related functions designed to achieve the general function
P	The appearance and spatial location of those components	The appearance and spatial location of the user interface objects

Table 4. Minimal set of attributes to be included in a classification scheme

Group	Attribute	Description	Required or Optional?
General	Identifier	Identifier for the usability problem	Required
	Overview	General explanation of the usability problem	Required
Context of Use	Interface object	Interface objects related to the usability problem (e.g., button, form, menu, etc.)	Required
	User group	User groups related to the usability problem (e.g., novice unfamiliar to certain functions, user groups with particular cultural background, etc.)	Required
	System function or task	System functions or tasks related to the usability problem (e.g., navigation, error prevention, form filling, etc.)	Required
	Organizational factor	Organizational factors related to the usability problem (e.g., usability problems related to communication, etc.)	Required
	Situational factor	Situational factors related to the usability problem (e.g., tasks with time limitation, etc.)	Required
	Cognitive stages of task	User's cognitive processing stages in conducting a task (e.g., goal formulation, establishing plan, state identification, etc.)	Optional
Design Knowledge	Interface design principle	Interface design principles related to the usability problem (e.g., consistency, visibility, error recovery, etc.)	Optional
	AH of design knowledge	An abstraction level of AH that is judged to be related to the usability problem	Optional
	Transformation in AH	Transformation in AH that is judged to be related to the usability problem (e.g., FP -> GF; GF -> PF)	Optional
Design Activity	Design activity in FBS	Design activities in FBS that are judged to be related to the usability problem (e.g., implementation, evaluation, etc.)	Optional
	Relevant activities in UCD	Relevant design activities in UCD that are judged to be related to the usability problem (e.g., task analysis, navigation design, requirements validation, etc.)	Optional

Table 5. Relationships between FBS model-based design process and usability-related design activities

Transformation in AH	**Relevant Process in FBS**	**Activity in FBS**	**Relevant Activities in User-Centred Design**
FP -> AF -> GF	Formulation	F -> Be	User and task analysis Analysis of context of use Usability requirements specification
GF -> PF -> P	Synthesis	Be -> S	Interface specification Metaphor design Dialogue design Navigation design Presentation design Error prevention and recovery design
P -> PF	Analysis	S -> Bs	Testing Review activities
PF <-> GF	Evaluation	Be <-> Bs	Requirements validation Usability evaluation
GF -> P	Documentation	S -> D	User manual design Help contents design
P <-> P	Structural Reformulation	S -> S	Refinement of design and code Fixing defects in design and code
PF <-> PF	Behavioural Reformulation	S -> Be	Requirements change
GF <-> GF	Functional Reformulation	S -> F	Change in users’ needs

Starting Points	5 Primary Categories	Sub-Categories	
Artefact Component	Visualness	Object (Screen) Layout	
		Object Appearance	
		Object Movement	
		Presentation of Information/Results	
		Non-Message Feedback	
	Language	Naming/Labeling	
		Other Wording	Feedback Messages
			Error Messages
			Other System Messages
			On-Screen Text
			User-Requested Information/Results
	Manipulation	Cognitive Aspects	Visual Cues
			Direct Manipulation
		Physical Aspects	
Task Component	Task-Mapping	Interaction	
		Navigation	
		Functionality	
	Task-Facilitation	Alternatives	
		Task/Function Automation	
		User Action Reversal	
		Keeping the User Task on Track	

Fig. 1. Classification of usability problems in UPT (Keehan et al. 1999)

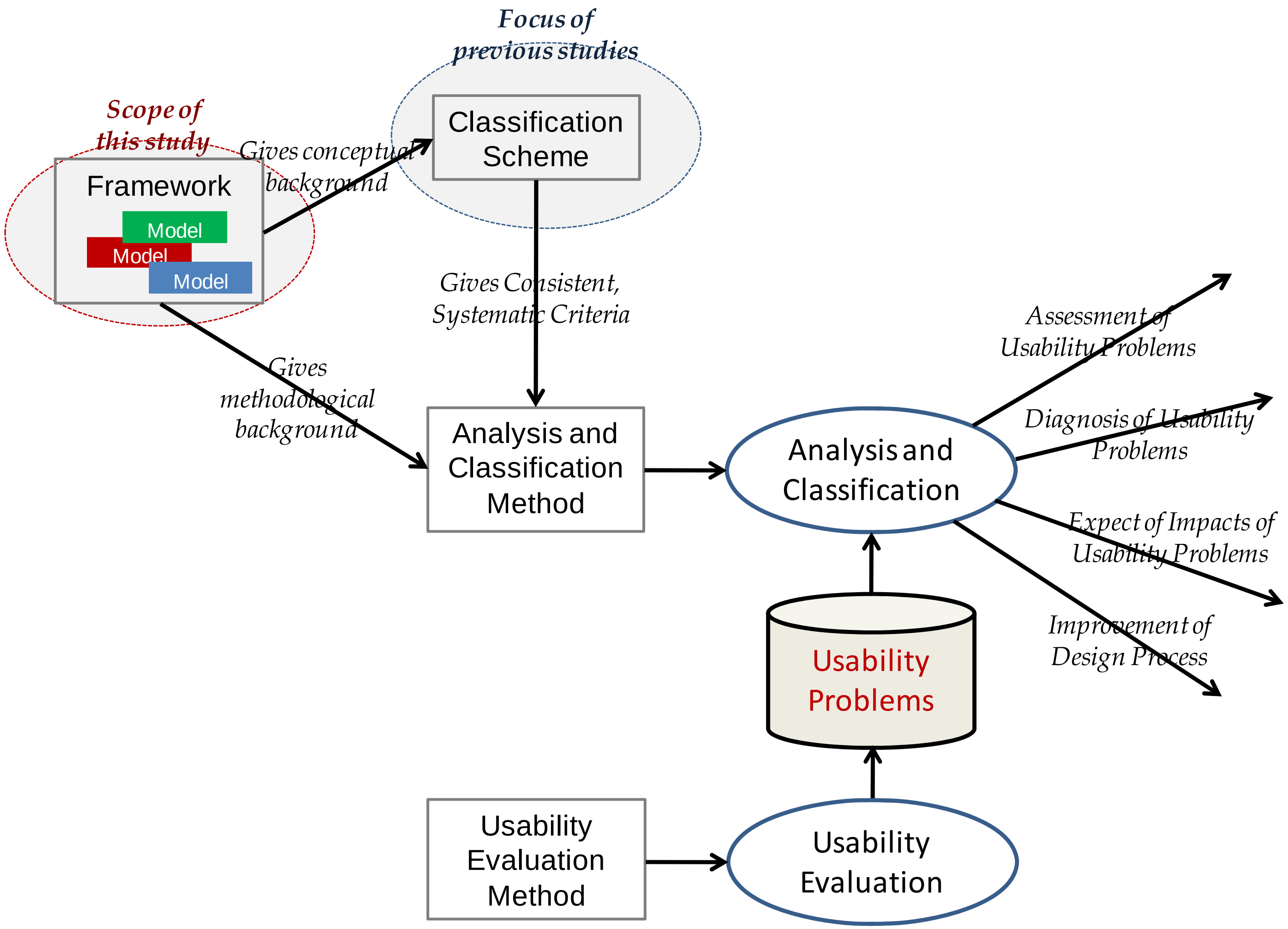


Fig. 2. Research method for developing a new classification scheme and conceptual framework

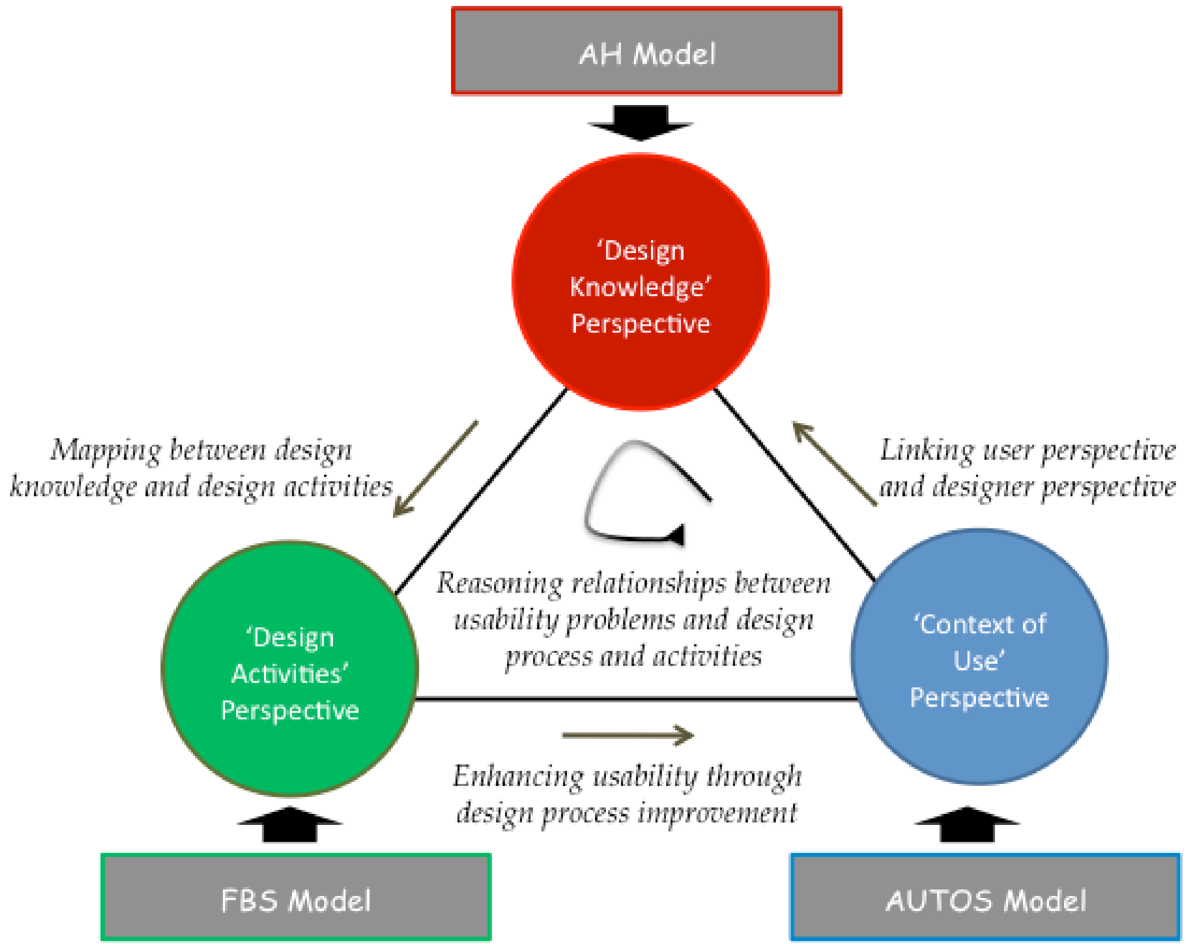


Fig. 3. Proposed conceptual framework

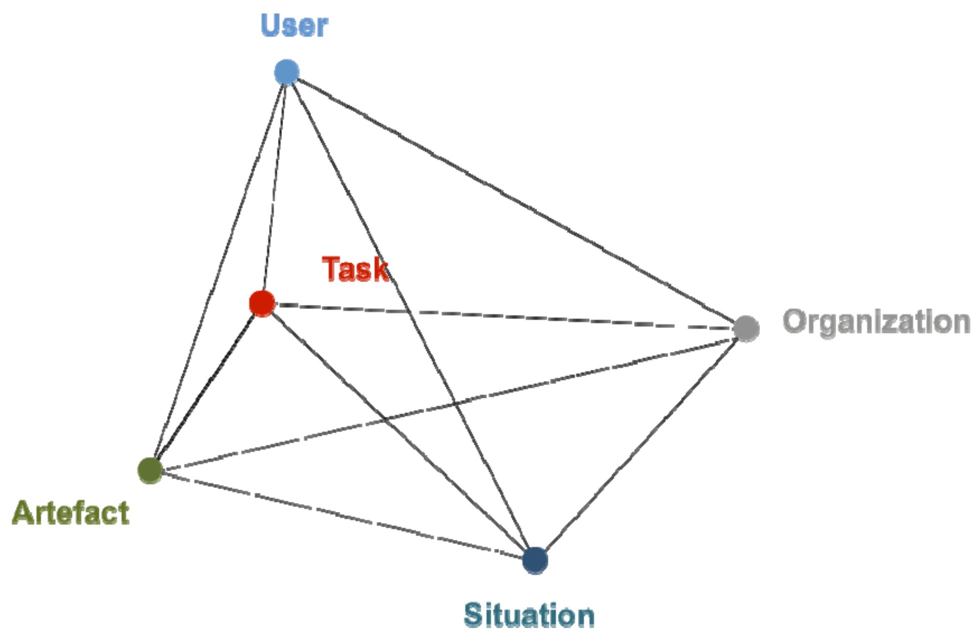


Fig. 4. AUTOS model (adapted from (Boy 1998))

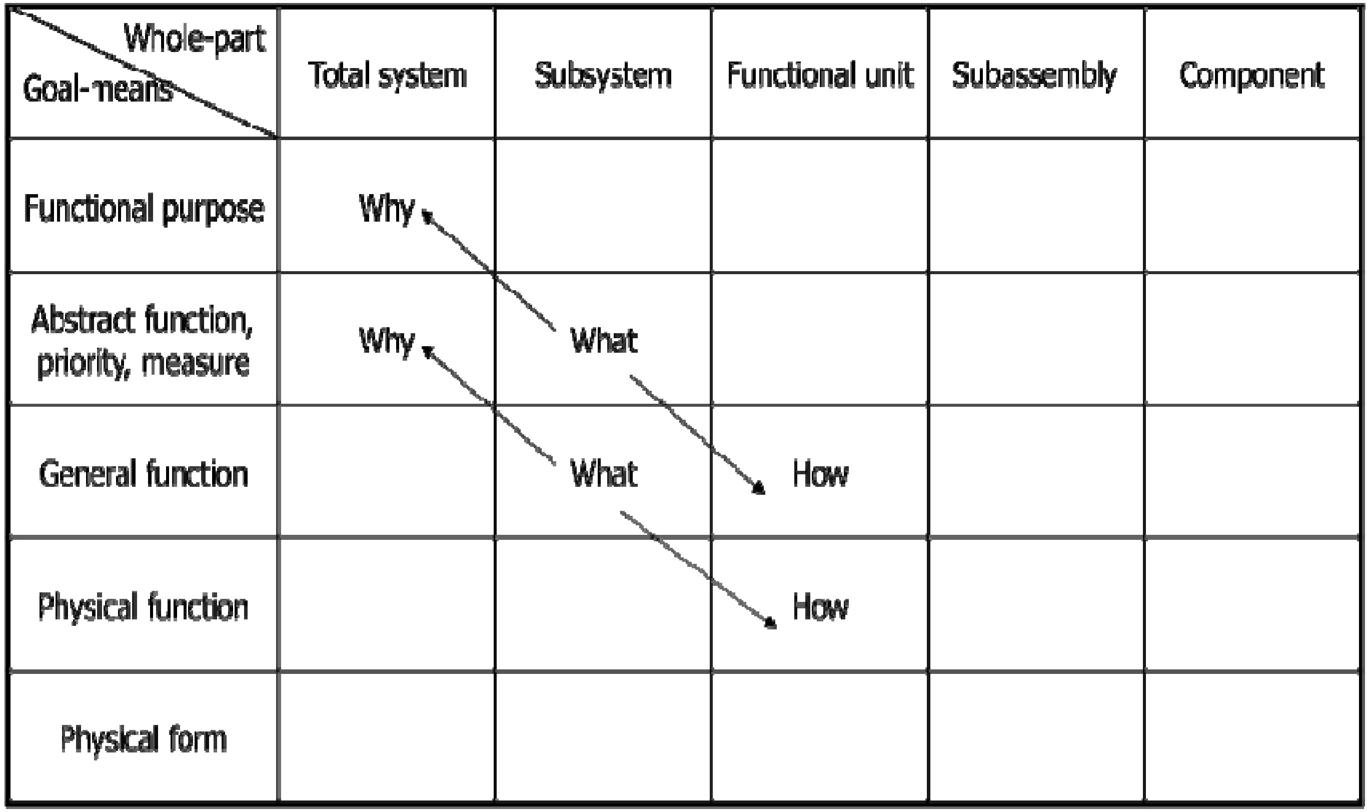
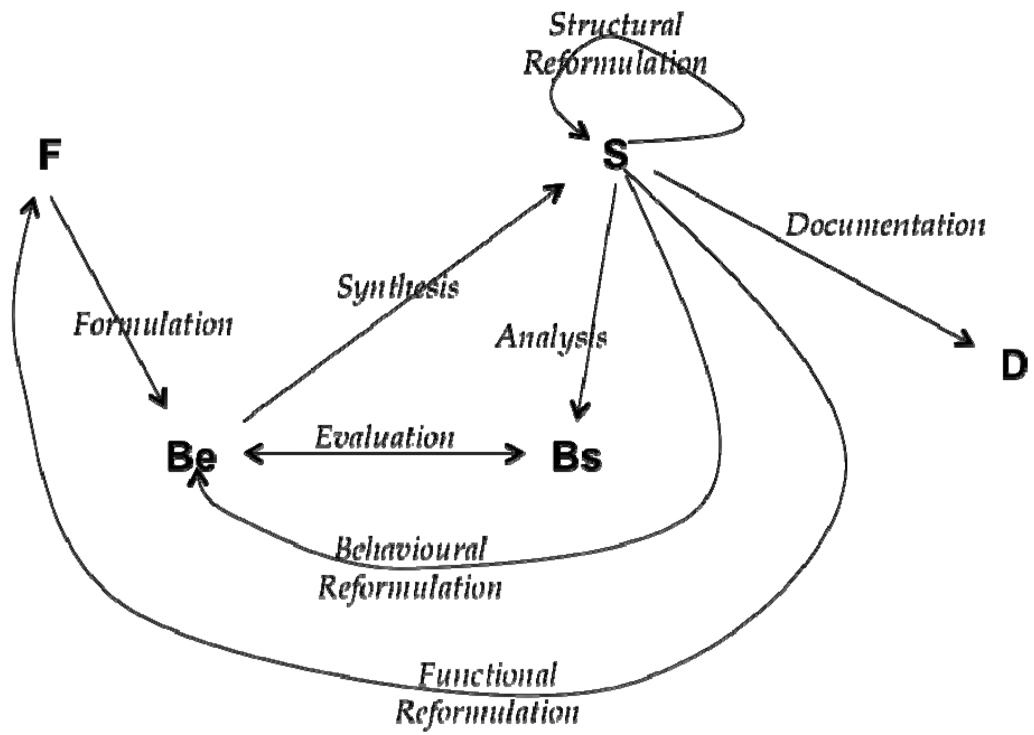


Fig. 5. Abstraction hierarchy of a system (adapted from (Rasmussen 1985))



F: Functions;
S: Structure;
Be: Expected Behaviours;
Bs: Set of Behaviours;
D: Design Description

Fig. 6. FBS model (adapted from (Gero and Kannengiesser 2004))