The Architecture and Development of Multi-Role Course Design of a Web-Based Group Training System

Xi Guo, Seng Chong, Sue Dyson, and Lorenzo Picinali

***Abstract*—Due to the fast development of information technology there are new opportunities for traditional training systems. Many e-training systems are proposed and implemented, however, there is rarely any research on group based e-training system that allow people to train in a group that involves different roles. This paper proposes an architecture for a web based surgery group training system that supports multi-role group training courses. A course model based on XPDL (XML Process Definition Language) is introduced to bridge the understanding of a graphical multi-role course design and the computer application. There is also an explanation on how the multi-role courses are designed and used in applications. Finally a demonstration experiment is given.**

***Index Terms*—E-training, group training, multi-role, web-based, XPDL, BPMN.**

I. Introduction

E-training which takes advantage of improvements in the internet surpasses the limitation of traditional education. Training processes are no longer limited by time or location [1]-[7]. However, most e- training methods only assist students to self-learning or self-practicing as an individual [5]-[6]. There is rarely research on group training. Group training normally includes processes with multiple roles, for example, during surgery training, doctors and nurses need to be trained to understand their roles so that they can cooperate as a team. In real life, this type of training is usually limited by the time, location and availability of members in the group. To solve this problem, a web based group training system is proposed in this paper. This system supports teachers to design online group training courses with multiple roles, and the system allows demonstration of the same training process according to different roles. To achieve the goals, there are three major problems that exist:

- Firstly how does the system represent multi-role based group training materials in a way both the designer and computer can understand?
- Secondly, how does the system support further modification of the courses according to personalised designs?
- Thirdly how does this system demonstrate the same process according to different roles?

Considering these issues, this paper proposes a method based on XPDL to bridge the understanding between human design and computer application. Subsequently, the architecture and a course model are designed to support flexible modification of the designed courses. To verify the design, an application is implemented to demonstrate the multi-role course design and demonstration. The results show that the proposed methods can be applied to solve the problems associated with a group training system. Finally, the conclusion and possible future direction are given.

Manuscript received October 9, 2012; revised December 13, 2012.

Xi Guo is with the Faculty of Technology, De Montfort University, Leicester, UK (email: xguo@dmu.ac.uk).

II. Literature Review

There are few tools for designing group training materials. A few projects have been carried out aiming at assisting group training and learning [1]-[4]. Second life is a project that uses 3D technology to construct an online virtual world [1]. A virtual hospital is built within second life to assist student understanding of a real hospital environment. However, this project does not provide any tools for creating training scenarios. [2] Some e-learning platforms such as Blackboard and Moodle are focused only on training material management [7].

Most designs are limited by a fixed practice session. Some companies design online surgery games using adobe flash, for example, "heart surgery game" which is published by Spil Games. Although it is not intended for educational purposes, it is very popular and display some general skills and processes of the heart surgery procedure. There is some commercial training software for medical students, such as the medical training system from SIMTICS Ltd and Surgicalskills from BMJ Group. Both products offer study materials such as text explanation, video, and 3D models. SIMTICS offers flash based practical sessions. However, all of the materials are pre-set and cannot be changed, there is also no functionality for separate different roles.

BPMN (Business Process Model and Notation) was proposed in 2006 by OMG (Object Management Group) [8, 9]. Using rich notation sets, BPMN offers a representation of the multitude of business scenarios representation standards to represent the communications and events between different roles. [8] There are already a significant number of design tools developed for BPMN such as TIBCO Business Studio. XPDL (XML based Process Description Language) [10] is a standard language used to describe BPMN (Graphical notation for drawing a Business Process). The description includes both logic content and graphical content of the BPMN. Using the description, automated computer tools can be used to analyse the BPMN models that are designed by different people. Therefore, BPMN is potentially a suitable representation for a multi-role based process. Meanwhile, XPDL can be used to bridge the understanding gaps between graphic based process design and automated

computer applications.

III. System Architecture Design

A. Course Model

As discussed in the literature review, BPMN can be used to represent a multi-role based process. However, most of the BPMN software only offers tools to generate and modify the diagram itself. A training process is usually attached with teaching materials such as pictures, texts and videos to explain real scenarios. The first problem is how can we use these materials together with the BPMN model and display the process according to the users' personalised settings?

XPDL provides a set of XML based language and format to describe the BPMN diagrams. By decoding XPDL, the computer will load he activity information as well as how these activities are linked together. By attaching extra text/image/video information to the target XPDL elements, it is possible to combine the process information with personalised information from users. Therefore, a course model with flexible structure needs to be introduced. In the current design, the course model is based on the XPDL model but with more personalised elements attached.

In the proposed design, there are seven elements that are used to describe the information. Some of these elements come from the original BPMN and XPDL definition [8]-[10] As shown in the Fig. 1, the details of these elements are listed as follows:

Pool: A Pool represents a participant in a Process. [8]-[10] In the course model it represents a container for partitioning a set of activities from other pools.

Lane: A Lane is a sub-partition within a Pool and in the course model. It is used to organise and categorise activities according to different participants. [8]-[10]

Activity: an activity may represent an event. An event is something that "happens" during the course process. There are three types of Events, based on when they affect the flow: Start, Intermediate, and End. [8]-[10]

Connector: Activities are related to one another via sequence flows. The sequence flow that connects different activities is called a connector. [8]-[10]

Text: a free-style text based materials that can be attached to the activity. It can be used to describe the activity details.

Image: an image that is attached to the activity. It can be used to demonstrate important content related to the activity.

Video: a video that is attached to the activity. It can be used to demonstrate important information that explains the activity.

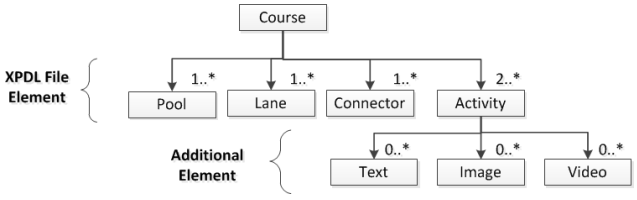

Fig. 1. Course model diagram

Some tools can be used to design the BPMN diagrams and translate the diagrams to XPDL files. The course model actually uses an object oriented method to describe the process based on the XPDL. A course is considered to consist of a set of basic elements that construct the course's basic structure. These basic elements ("Pool", "Lane", "Connector" and "Activity") are shown in the first layer of course model in Fig. 1. The basic elements can be found from corresponding nodes in XPDL files. An activity further consists of a set of multimedia elements ("Text", "Image" and "Video"). These multimedia elements located in the second layer in a course model in Fig. 1. The resources can be modified by the users later when designing the course. The information in the XPDL is under corresponding XML nodes with the same name. "1..*" represents the fact that each course will have at least one element. For example, each course will have at least one Pool and one Lane element. Using this structure, the course model contains enough information and a flexible structure for the computer to process. A further application for course design, modification and demonstration can be built based on the course model.

B. Cooperating Course Designing

In fact, the most important reason to design the model is to make it possible to share and reuse the knowledge and to make course design more flexible and easier for the designers. Usually, the course designers need to design from the process to the detail multimedia materials, which is a significant amount of work. Also, the work cannot be reused easily because everything is linked together. The course model makes the course structure flexible enough to incorporate information from different sources. That is, the course generation process can be finished by different designers, which makes it possible to reuse the knowledge and makes the design simpler.

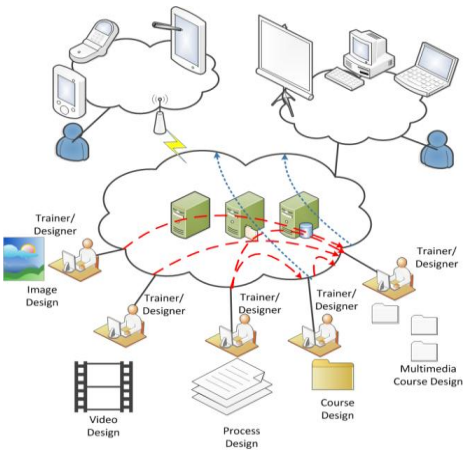

Fig. 2. Cooperation design and multiple user access design of the platform

Fig. 2 explains how to achieve the knowledge sharing and reuse. The whole system is built upon the internet and in the middle the three server icons represent the system that stores all the applications, course data and user data. At the top of the figure, there are two groups of users. One group of users represents users accessing the system via traditional tools like computer, laptop and monitors. Another group of users access the system via mobile devices such as mobile phones, tablets and PDAs.

At the bottom of the figure, there are other types of users. There are the different designers:

- **Image designer** designs image materials for the course.
- **Video designers** design videos such as demonstration of the surgery process.
- **Process designers** design the basic course process using BPMN/XPDL tools. Finished XPDL files can be uploaded to the server.
- **Course designer** design how to describe the course in a text based way.
- **The multimedia course** designers reuse and modify the course from other course designers. Multimedia information such as images and videos available can be added to the course activities.

The long dash line marks the information flow among different designers. All the media materials are uploaded to the internet. For example, the image designer may upload some photographs and process designer can upload XPDL process description. Therefore, the only thing the course designer needs to do is reuse the XPDL process and focus on is adding these materials for activities in the course. Moreover, these designed courses can be further modified by other course designers. The key point is that the knowledge can be shared and reused, and compared with other designs cooperation can make the course design significantly easier and more flexible.

C. System Architecture

To achieve the design, the intention is to build basic function modules which allow various web applications developed upon. Behind the functions there are detailed functional logics as well as a data management system that need to be designed. Based on the functional requirements, the proposed system architecture is shown in Fig. 3. The modules are designed to be function across five layers, which are divided according to their roles in the system. There are standard APIs for communication between each layer. Each layer has the following functions:

User Application Layer: this layer offers web based applications with User Interface (UI) that allows users to access the system via an internet browser.

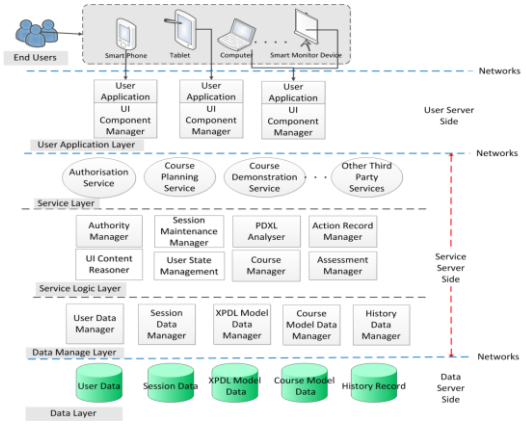

Fig. 3. System architecture design

Service Layer: to support the web applications, there are backend applications deployed on the server. This layer offers different services to support user web applications on the User Application Layer. For example, it provides "Course Planning Service" for application to help users design personalised courses. The "Authorisation Service" supports security and allows users to log on to the system. "Course Demonstration Service" offers functions to demonstrate the course design result.

Service Logic Layer: The Service Logic Layer is designed to achieve and provide commonly used functions to support the higher level services in the Service Layer. Authority Manager offers functionality to authenticate users . UI Content Reasoner generates a dynamic user interface according to user's requests by deciding the content that should be displayed to the user via the interface. The Session Maintenance Manager provides functions to maintain and track the users' state when they access the system. User State Management provides functions to update and synchronise the user's current state during group training and with other members. XPDL Analyser accesses and analyses the input XPDL files, collects and stores the required elements'. Course Manager provides functions to modify course models, such as add, update and delete a course model. Action Record Manager offers functions to record and tracking user actions during the group training process. Assessment Manager provides functions to assess users' behaviour during the group training process.

Data Management Layer: This layer provides most of the functions needed to access the data sources. To avoid read and write conflicts, this layer offers five modules for five data sources stored in the Data Layer. There are some commonly used functions that can be used by modules that need to access the data source, such as add, update, delete and search which is available in each module. The User Data Manager is responsible for user related data such as user name and password. The Session Data Manager is responsible for data related to a practice session. The XPDL Model Manager is in charge of the important elements' data from XPDL, such as pool, lane and activity. The Course Model Data Manager is in charge of the course data.

Data Layer: as shown in the figure, in this layer, there are several different data sources. Each source could be a set of databases or other forms of data storage such as XML files. Currently, most of the data is stored in database. User Data stores information related users and is controlled by the User Data Manager. Session Data is controlled by the Session Data Manager. XPDL Model Data is operated by the XPDL Model Data Manager. Course Model Data is controlled by the Course Model Data Manager and History Record is controlled by the History Data Manager.

The diagram also shows the location of the components. There are three kinds of server (User Server, Service Server and Data Server) that carry out different jobs in this project. The servers can be within the same internal networks or they can be located in different networks connected by the internet. The key aspect of this structure is the applications in the system are loosely coupled so that any module changes won't influence the whole system much.

IV. Multi-Role Course Design Process

The architecture shows how the functions are deployed within the whole structure. However, there is still significant work required to make the modules work together to support multi-role course design. There are many cases in various

surgery processes that require teamwork. For example, carpal tunnel surgery involves several different roles: doctor, nurse and patient. The question is how to generate such a course that can relate the different roles and add the multimedia data for the required activities. This section will explain how the current system uses a course model to solve this problem. There are three steps in the system for a multi-role course generation and modification.

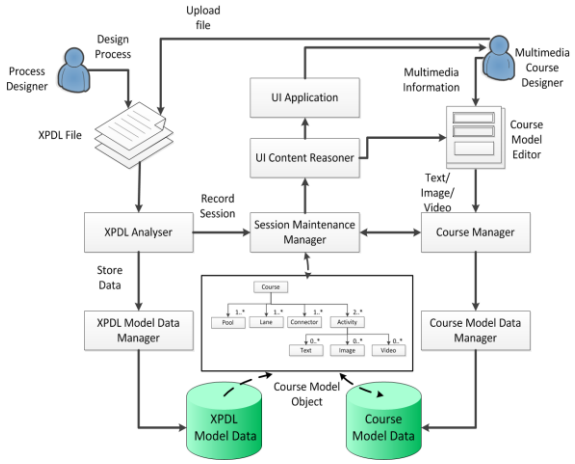

Fig. 4. Generation and use of course model

A. Step One: Get Element Data from XPDL File and Initialise Course Model Object

Fig. 4 shows how the related modules in the architecture work together to generate and use the course model. Two users are involved in the process. "Process designer" designs the process using the BPMN tools and generate XPDL files for other users to access. In our project, "Studio for Designers" in TIBCO Business Studio 3.5 Community is used in the demo to design the process. The rest of the task is finished by the "Multimedia Course Designer".

After the XPDL file is uploaded, the XPDL Analyser decodes the file and collects the important elements. The important elements include "pool", "activity", "connector" and "lane". In XPDL, there are important attributes attached to these elements which also need to be been collected. For example, in "activities", there is the activity name, display text and position of the activity block. After that the XPDL Analyser stores the data in the XPDL Model data via XPDL Model Data Manager. At the same time, the upload XPDL information is reported to Session Maintenance Manager and the system generates a reference ID for the information. When the user selects this XPDL reference ID, the Session Maintenance Manager then invokes the course manager to get XPDL data from the database. When the user chooses to generate a new course, the XPDL data can be put in a Course Model Object which is designed according to the course model described in the section before.

B. Step Two: Modify the Course Model

To make it easier for users to modify the course, the Course Model Object content is transferred to a UI Application by UI Content Reasoner. The user can select target activities through the UI Application. The Course Model Editor is an interface of the UI Application for users to upload text, picture and video to the selected activity. Once the user uploads the information to a selected activity Course Manager obtains the Course Model Object via Session Maintenance Manager. According to selected activity ID, the Course Manager will attach the uploaded information to the corresponding activities under the Course Model Object. In this way, the users can attach personalised information to any activity blocks in the diagram. After editing, the information will be stored in the Course Model Data and can be used in other applications in the future.

C. Step Three: Build Up UI Application Using Course Model

Finally, to make sure the course is correctly designed, the system offers a function to demonstrate the course effect. When the user requests demonstration of the course through the UI Application, the requests first arrive at the Session Maintenance Manager, and then this module obtains the corresponding Course Model Object from the data base. The object will then be used by UI Content Reasoner to generate a formatted message. The message will be sent to the UI Application on the client side. The UI Application then decides how to use this information and how to display the demonstration.

V. Experiment Result and Discussions

A demonstration application is implemented using Java, HTML, JavaScript and flash flex to achieve the course design and modification and demonstration. The experiment is processed based on following functional criteria:

- The system should be able to load designed multi-role courses correctly;
- The course should be able to be modified and the result can be updated to the system correctly;
- Personalised information can be added in to the course material correctly;
- The modified courses can be demonstrated correctly according to selected roles.

The results are shown in Fig. 5. "Course Management" allows users to modify the information for each activity. "Simulation Management" (Fig. 6) allows users to simulate the program. In "Course Management" panel, in the middle a BPMN diagram shows the carpal tunnel course diagram. When the user clicks the activity block in the diagram, corresponding information will be loaded in the form below the diagram. Fig. 5 shows when "Start Event" in Nurse is selected the user added a simple text introduction ("This is a start event. The Nurse gets ready for the surgery."), uploaded a diagram called "carpalTunnel 1.jpg" reference from "SafeComputingTips.com" and a video called "Carpal tunnel syndrome" from YouTube. When the user clicks the "Simulation" button, the control page will be changed to the "Simulation Management" panel. On the left side, a list of panels shows the current activity names for all roles available in the process. On the right side, there is an activity display panel showing the selected role activity details. In Fig. 6, the user selected "nurse" from the top selection box. The role name and introduction of the activity is shown on the activity display panel. The panel also shows the related picture and video for the selected roles. The web application implements an algorithm detecting the current activities for each role. By clicking "Next Step" and "Former Step" buttons, the users

are able to browse the required activity according to the process. For example, in Fig. 6, if "Next Step" is clicked, the activity will move to the next activity "Get patient request". The content of the activity display panel will be updated accordingly.

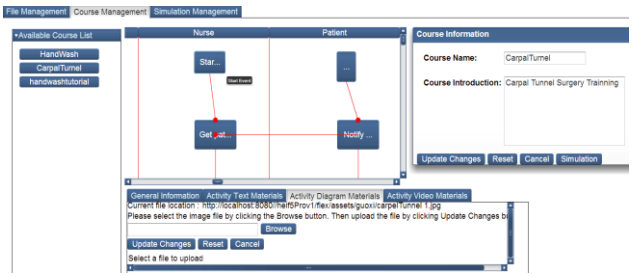

Fig. 5. Carpal tunnel course in "course management" panel

Fig. 6. Carpal tunnel course simulation

The result shows the design achieves all of the expected functional criteria. The course model designed using BPMN diagram can be correctly loaded, modified and demonstrated by the system. The personalised information can be added to the course activity correctly and later displayed during the demonstration. The design process can be reused in different courses and the course designer only needs to concentrate on organising course materials.

VI. Conclusion and Future Work

This paper aims to propose a multi-role course design method of a web-based group training system. The system prototype has been presented to the project team and the partner surgical team, to verify the abilities of the system. The main contributions are:

- Proposed and implemented a XPDL based method so that diagram based multi-role process can be processed by the computer programs.
- Design and implement a web based multi-role course design system that allows the multi-role course to be modified and processed in different applications.
- Propose and implement the method that supports separating process design and multimedia materials design from a complicated course model. Therefore, multi-role course design becomes more flexible and convenient for designers.

However, the current work only supports simple BPMN design without any conditional divergence or convergence during the process. Further work is required to handle more complicated multi-role training with such conditions. Also, further design is required to support group simulation practice that allows multiple users to practice in the same course process.

References

[1] S. Kumar, M. Hedrick, C. Wiacek, and J. I. Messner, "Developing An Experienced-Based Design Review Application For Healthcare Facilities Using A 3D Game Engine," *Journal of Information Technology in Construction*, vol. 16, pp. 85, 2011.

[2] M. N. K. Boulos, L. Hetherington, and S. Wheeler, "Second Life: an overview of the potential of 3D virtual worlds in medical and health education," *Health Information and Libraries Journal*, vol. 24, no. 4, pp. 233–245, Dec. 2007.

[3] S. Freitas. (Nov. 2008). Serious virtual worlds: A Scoping Study. *Bristol: Joint Information Systems Committee*. [Online]. Available: http://www.jisc.ac.uk/media/documents/publications/seriousvirtualworldsv1.pdf

[4] W. L. Heinrichs, P. Youngblood, P. M. Harter, and P. Dev, "Simulation for team training and assessment: case studies of online training with virtual worlds," *World Journal of Surgery*, vol. 32, no. 2, pp. 161-170, 2008.

[5] E. E. Tam, F. Badrat, R. J. Marceau, M. A. Marin, and A. S. Malowany, "A Web-based virtual environment for operator training," *IEEE Transactions on Power Systems*, vol. 14, pp. 802-808, August 1999.

[6] A. Martens, J. Bernauer, T. Illmann, and A. Seitz, "Docs 'n Drugs – The virtual policlinic. An Intelligent Tutoring System for Web-based and Case-oriented Training in Medicine," in *Proc. American Medical Informatics Conference*, Washington, USA, pp. 433-437, 2001.

[7] D. Dagger, A. O. Connor, S. Lawless, E. Walsh, and V. P. Wade, "Service-Oriented E-Learning Platforms: From Monolithic Systems to Flexible Services," *Internet Computing, IEEE*, vol. 11, vol. 3, pp. 28-35, 2007.

[8] OMG. (Jan. 2011). Business Process Model and Notation. [Online]. Available: http://www.omg.org/spec/BPMN/2.0/PDF.

[9] J. Recker, M. Indulska, M. Rosemann, and P. Green, "How Good is BPMN Really? Insights from Theory and Practice," in *Proc. 14th European Conference on Information Systems*, pp. 1582-1593, June 2006.

[10] WfMC. (Aug. 2012). Workflow Management Coalition Workflow Standard Process Definition Interface -- XML Process Definition Language. [Online]. Available: http://www.xpdl.org/standards/xpdl-2.2/XPDL%202.2%20(2012-08-30).pdf

Xi Guo was born in China in 1982. She graduated from Beijing University of Post and Telecommunications with the Bachelors and Masters degrees in Computer Science in 2004 and 2007 respectively. She then obtained her PhD degree from Loughborough University in 2011. She is currently a research assistant at De Montfort University for the 'Feasibility Study of a Collaborative Virtual Learning Environment for Surgical Teams' project. Her main research interests are artificial intelligence, internet of things, e-learning, dynamic web user interface and web services.

Seng Chong gained his MSc in Mechatronics with Distinction, and later obtained his PhD degree at De Montfort University. He is currently a Senior Lecturer within the School of Engineering, Media & Sustainable Development of the Faculty of Technology. He is also the Programme Leader for the MSc Mechatronics course. His main research interests are Virtual Engineering, Machine design and control, Connected home technologies and system, Product life cycle information, Model driven architectures and Model-based design, and Creative Technologies.

Sue Dyson was born in Staffordshire, UK, in 1960. She is a nurse and midwife by professional background being educated at Wolverhampton University where she obtained an honours degree in Education Studies in Nursing, followed by a Master's degree in Health Policy and Organisation at University of Nottingham, and a Doctorate in Education at the University of Leicester. Sue is

currently Reader in Nurse Education within the School of Nursing and Midwifery at De Montfort University. Her main research interests are pedagogy in nursing and midwifery, transcultural nursing, educational experiences of children with sickle cell disease, and volunteerism among health care professionals.

Picinali was born in Milano, in 1981. He graduated from the Computer Science Department of the Universit à degli Studio di Milano in 2005, and received a Ph.D. in Music Technology from De Montfort University in 2010. He is a Senior Lecturer in Music/Audio Technology in the Department of Media Technology at De Montfort University and an active researcher in the Interactive and Media Technologies (IMT) and in the Virtual Reality and Assisted Living (ViR.AL) research groups. Currently, his main research interest is in the use of virtual reality interactive applications for hearing and visually impaired individuals.

Joseph J. Dias (MD, FRCS (Edinburgh), FRCS (England), MBBS) is currently a practicing orthopaedic surgeon at the University Hospitals of Leicester in the UK and the head of the University of Leicester Orthopaedic Division. He has research interests in wrist disorders, assessment, nerve recovery, education and organisational change.