[PoESKrSt00] *Nathalie Poerwantoro, Abdulmotaleb El Saddik, Bernd Krämer, Ralf Steinmetz;* **Multibook's Test Environment;** Proceedings of the 22nd International Conference on Software Engineering (ICSE2000), Juni 2000, Limerick, Irland, S.

In Proccedings of the 22nd. International Conference on Software Engineering (ICSE2000), June 4 -11 2000, Limerick, Ireland

Multibook's Test Environment

Nathalie Poerwantoro[1], Abdulmotaleb El Saddik[2], Bernd Krämer[1], Ralf Steinmetz[2]

1
FernUniversität Hagen
Feithstrasse 140
58084 Hagen
Germany
49-2331-9874541
{Nathalie.Poerwantoro, Bernd.Kraemer}@fernuni-hagen.de

2
Darmstadt University of Technology
Merckstr. 25
64283 Darmstadt
Germany
+49 6151 166158
{Abdulmotaleb.El-Saddik, Ralf.Steinmetz}@kom.tu-darmstadt.de

ABSTRACT
Well engineered Web based courseware and exercises provide flexibility and added value to the students, which goes beyond the traditional text book or CD-ROM based courses. The Multibook project explores the boundaries of customized learning materials by composing learning trails dynamically as learners have set their profile to access a course. In this paper we first give an overview of the core project ideas and illustrate them along our Software Engineering course. Then we present a novel extension to the project's exercise environment with a graph editing component that particularly fits the needs of structure-related assignments.

Keywords
Online Courseware, user profile, dynamic composition, self assessment

1 INTRODUCTION

There is a lot of well developed educational software with a large potential for online learning. As individual programs, they often represent excellent applications of sophisticated computing capabilities and a good contribution to the library of educational software. However, they lack cohesion as an organized collection because every software application is vertically engineered to comply with its specific domain.

To achieve a horizontal solution, a more general approach needs to be defined. Our model takes a user-centred view. That is, we ask ourselves what students want rather than what content developers prefer to offer. The tools developed in the Multibook project enable content authors to employ a multilayered presentation to improve the effectiveness of learning. It also employs reusable components, which can be combined to create complete lessons for a given subject. This can be achieved by describing these components with adequate metadata. We rely on the Learning Object Metadata Standard (http://ltsc.ieee.org/wg12/). In section 2 we give an overview of the Multibook project. Section 3 presents its corresponding self assessment tool.

2 MULTIBOOK

To improve searching, composition and reuse of knowledge components, the Multibook project [1], which involves the Technical University of Darmstadt and the FernUniversität Hagen, aims to create a system providing an optimal support for the learner. It integrates a variety of different learning styles and a Java XML-based Exercise and Environment, which can be used to test students' skills remotely.

As the research area of adaptive hypermedia systems is being explored for a long time now, it is the particular goal of the Multibook to examine how *multimedia* integrating text, images, audio, video, and animations can be used best to support the learning process. This approach requires the knowledge about which of these media have to be used with respect to the particular content to be explained. Especially the meaningful use of different media is being neglected in many commercial systems following the paradigm "the more – the better".

The aim of Multibook is to have individual views on the learning material according to the needs and preferences of individual users. This is achieved with the help of the user information stored in a user profile. According to this profile, a table of contents is created and a lesson is generated on the fly. For example, a student learning for an exam will probably be interested in learning the broader terms, the components and the applications of the topics she is going to be asked in her exam. A businessman or a

manager on the other hand, who has different knowledge prerequisites depending on her vocational experiences, is interested in different aspects of the same topic such as implemented systems, economical aspects and alternatives.

For our software engineering example, some of our students who participate in the course for continuing education purposes may have a decent knowledge about the specific characteristics and challenges of building distributed software systems. They just want to know more about the role and functionality of CORBA at this point. They would provide information about their pre-knowledge while setting their profile. Multibook would drop all learning components referring to basic concepts such as naming, resource contention, synchronization, or trading and compile a table of contents focusing on CORBA based application development.

One of the aspects we face today is the use of applets in traditional hypertext systems as well as in Intelligent Tutoring Systems. Java applets are employed very often to explain complex processes to deepen the understanding of difficult parts of educational documents. However, the use of Java applets does not have to be restricted to simulations or visualizations of specific problems. Many examples show that applets can be used for exercises and tests and therefore offer more flexibility than other media. Another advantage of applets is the possibility to use parameters so that they can be used for different learners and preferences [3].

Another aspect we are dealing with, is an exercise environment that lets students apply and experience to pre-defined problems or problems of their choice what they have learned. Thereby we expect to raise the attractiveness of the learning material and enhance the students' motivation, especially that of distance learners.

In the design of these environments we have attempted to associate learning objectives with appropriate multimedia solutions with respect to different teaching functions. They include motivation, presentation of new structures, practicing new behavior, modeling, exploration by trial use and error and application of new techniques. To achieve a higher degree of independence from proprietary formats and platforms, ease of use and portability, we decided to rely on HTML, XML and Java as core technologies for the implementation of the exercise environments.

3 SELF ASSESSMENT TOOL

Applets are often used by educators within Web-based interactive exercises. Since most applets are programmed to cover a specific topic, it is not very flexible to update the questions embedded in the applets, especially when the authors wish to continually add new exercise questions. To solve this problem, we created an exercise environment consisting of tools that help authors with creating and adding new questions. We also developed an exercise environment applet that can pick a random set of questions from a question pool, adapts them to the student's preferences and conforms to exercise specifications defined by the author. The Extensible Markup Language (XML) is used to describe the question and the exercise as a whole. Internally each exercise question is described in XML and later parsed by the exercise environment applet. The exercise environment applet presents the question in an appropriate format depending on the question attributes and its companion media. That way a large question pool can be generated more easily. The configuration of an exercise, which describes how many sections and questions exist and what the type of questions **in each** section should be, is also described in XML. It is also mapped into an exercise specification object used by the exercise environment applet at the beginning of an exercise session. This approach provides flexibility to authors (Fig. 1), since questions for the exercise can be added any time to the questions pool together with companion animation, pictures or applets. The exercise environment [2] is divided into :

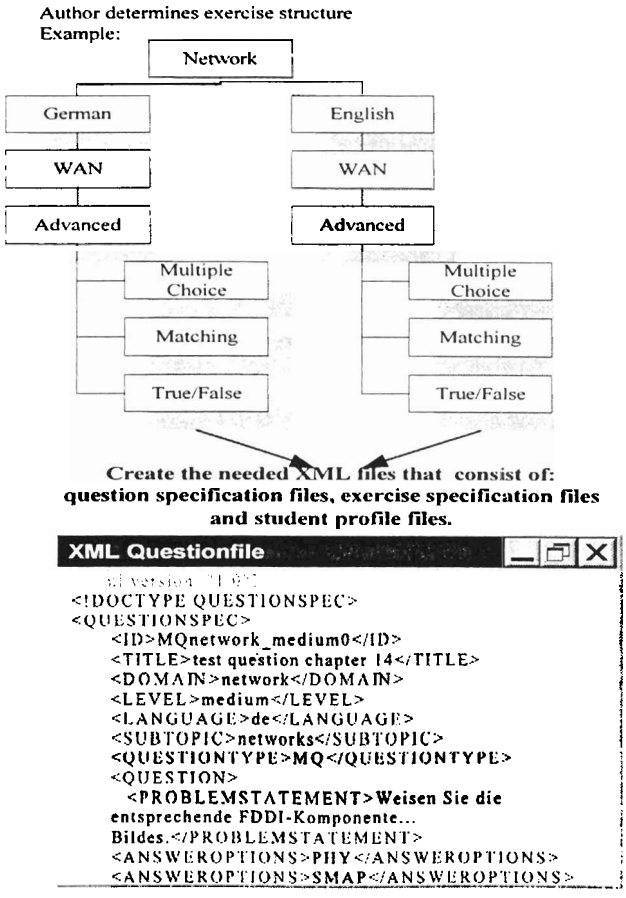


Figure 1: The steps taken by an author in preparing and creating an exercise for the exercise environment

- authoring tools : a question creator application that provides the tutors with an easy to use tool to add new questions into the question pool. An exercise configuration application is used to describe the exercise structure.

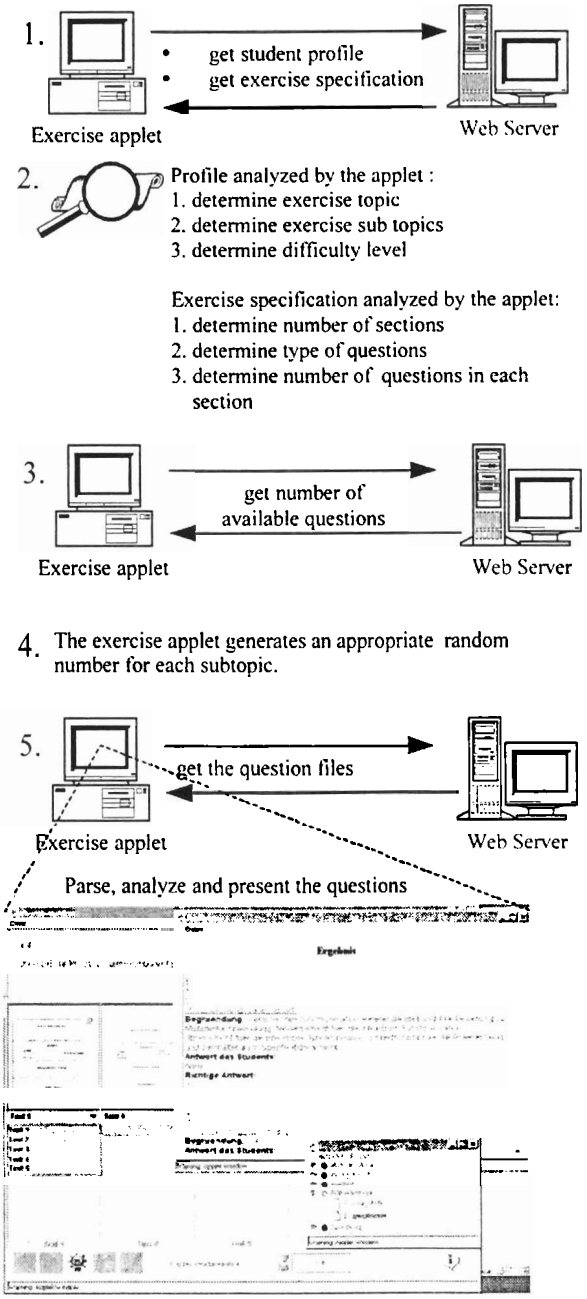


Figure 2: Steps taken by the exercise environment applet to setup an exercise session.

- exercise environment applet, which is the work environment for the students. A student profile is loaded by the applet at the beginning of an exercise session. The student's profile provides the information of selected exercise topic, subtopics, language and difficulty level. Then the selected exercise scenario is loaded, parsed and analyzed. The applet uses the collected information to prepare itself with the right exercise parameters (Fig. 2). The exercise environment then creates a random number for each question category to provide variation each time the exercise is restarted. The creation of random number is normalized within the interval of available questions.

To support exercises in software engineering education, an additional module allowing graph editing has been added recently. Graphical representations play a central role in software engineering. Use case, interaction and collaboration diagrams are often used in requirements engineering to document scenarios. Class diagrams, activity diagrams, sequence diagrams and other type of graphical notations help to visualize design aspects, software architectures, and behavior. This module provides an extension in the question variation and presentation (Fig. 3).

The module consists of:

- graph editor used by the author to describe the question and the corresponding solutions. Attributes for the nodes and edges include their position and relation to each other. With our exercise environment the author is creating an overlay graph whose nodes and edges are marked with two binary attributes "correct" and *"visible"*. Currently we provide a small collection of node and edge types the author can choose from through a menu. This collection can, of course be expanded as needs arise. This editor application produces two files: an XML question file and an applet.
- graph drawer applet. This applet will be activated during an exercise session. The applet presents the question graph and captures the student's inputs. Nodes and edges are presented as defined by their attributes. In some cases edges might be invisible to the students and must be created by the students. The graph drawer applet sends the answers back to the exercise environment applet.

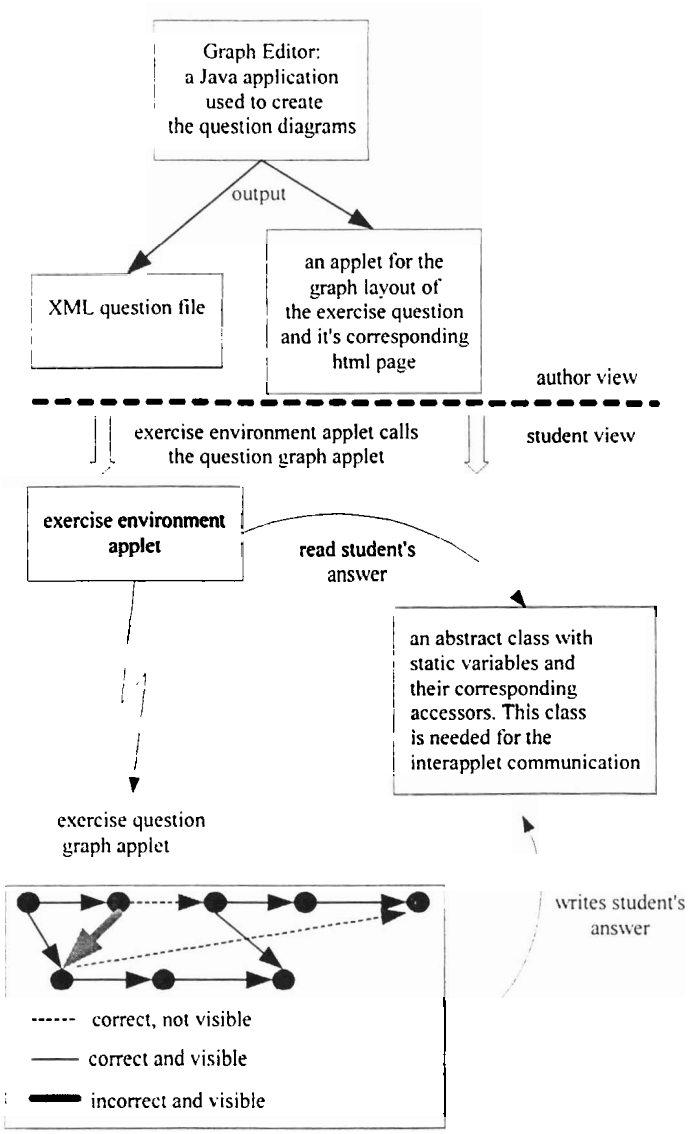
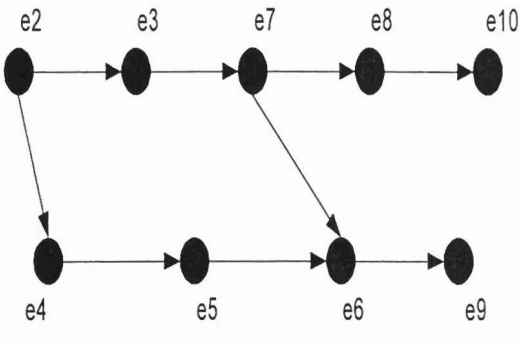
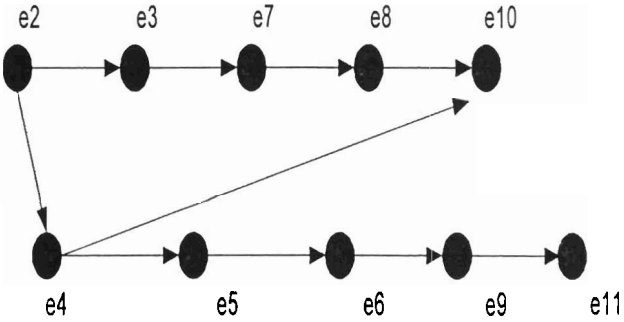


Figure 3: Concept of the question graph applet

In our course, for example, students have learned to specify non-sequential behavior in a declarative way. Partially ordered graphs are used to visualize such behavior for finite observations. In an assignment the student is given the process graph shown in Fig. 4 together with the question whether this process satisfies a given specification. In the process graph, the author has added one node and two edges that are not correct and has left out an edge that is required. The student's task is to verify these errors and manipulate the given graph to create the correct solution.



This diagram shows the correct answer



This shows the wrong graph that has to be corrected by the students

Figure 4: An exercise example

The first evaluation of this approach is planned through distributing CDs to the students together with a questionnaire covering several aspects such as usability, user interface, navigation, learning experience. Additional information of the students' equipment and previous experience with computers will also be collected and included into our questionnaire.

REFERENCES

- [1] Multibook, http://www.multibook.de
- [2] B. J. Krämer, N. Poerwantoro. An XML-based Approach for Web-based Self Assessment, ED-MEDIA 2000, Montreal, Quebec, Canada, June 2000
- [3] A. El-Saddik, S. Fischer, R. Steinmetz. "ITBeankit: An Educational Middleware Framework for Bridging Software Technology and Education". ED-MEDIA 2000, Montreal, Quebec, Canada, June 2000.