Adaptive Learning and Testing with Learning Objects

Jane Y-K Yau & Mike Joy
Department of Computer Science
University of Warwick
Coventry, UK
janeyau@dcs.warwick.ac.uk

Abstract: Learning objects are pedagogic software components which are interoperable, exchangeable and reusable between web-based learning environments, and adaptive learning and testing can provide each student with personalized learning content or assessment questions. In this paper, we describe our existing adaptive authoring tool which can be used to convert a non-adaptive course into an adaptive one which uses learning objects. Learning material can be reused in our framework which consists of lesson instructions, pre-tests, performance tests and proficiency tests. Our current metadata for describing the learning material will be merged with a simplified and customised version of *Learning Object Metadata* to allow the import and export of learning objects between different learning environments.

Keywords: Learning Objects, Adaptive learning, Adaptive testing

Introduction

Adaptive learning customises for each learner the selection and presentation of learning material according to their previous and current knowledge, learning aims and objectives, learning or cognitive styles and special needs (Hockemeyer & Dietrich , 2002). Similarly, adaptive testing can effectively assess and distinguish students of different abilities by dynamically adjusting its proficiency to suit all students (Wainer *et al.*, 2000).

A learning object (LO) is a digital learning resource which fulfils a single learning objective and can be reused in many different contexts (Mohan & Greer, 2003) and which can be constructed around *Lesson Instructions, Pre-Test, Performance Test* and *Proficiency Test* (ASTD & Smartforce, 2002). *Lesson Instructions* contain for example text, multimedia (graphics, animation and audio), questions and exercises. *Pre-Tests* assess students to ascertain their level of understanding in order to locate them at the appropriate level of instruction (Arroyo *et al.*, 2001). *Performance Test* consists of summative scored assessments which evaluate the student's ability to successfully accomplish a specific task, using the skills acquired. *Proficiency Test* evaluates whether the students have successfully assimilated the learning material and mastered the required skills.

As part of ongoing work in computer-assisted learning technologies (Joy *et al.*, 2002), an authoring tool called *Online Computer Teaching Aid* (OCTA) was developed to help students learn computer science course material. Its features are common to many computer based teaching aids however it aims to provide each student an adaptive learning experience based on their previous performance. This paper describes the enhanced OCTA software which fulfils the following two objectives:

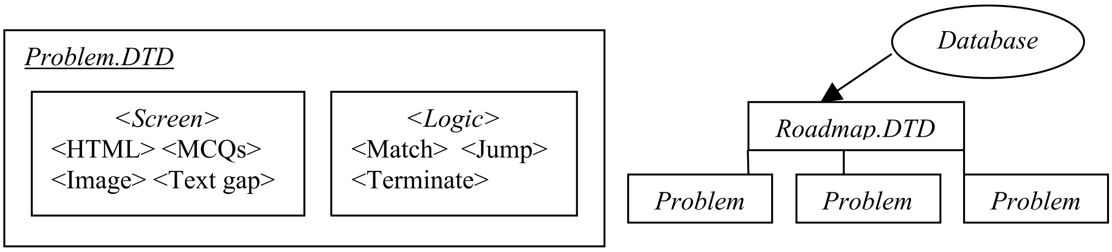
- An integration of re-usable LOs into adaptive learning and testing and
- Import/Export LOs to/from other learning environments.

System Overview

A non-adaptive Introductory Java programming course (Yau, 2002) aimed at students before they enter higher education has been imported into OCTA to demonstrate its adaptive features. Personalized content is desired because the target users may have different levels of programming experience. Ten topics have been imported; each assigned with an overall difficulty level ranging from 1 to 10 and each topic is further sub-divided into three proficiency levels (Novice, Intermediate and Advanced).

Figure 1 show the two entities contained in OCTA called *Problems* and *Roadmaps* which are represented as XML DTDs (Document Type Definitions) (W3C, 2003) and contain metadata tags.

Figure 1 Problems and Roadmaps.



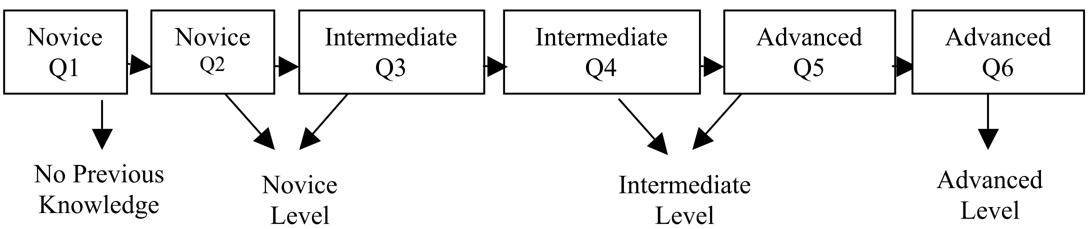
A Problem can be regarded as a LO because it primarily contains learning material and/or an individual question or a set of questions on a particular topic. They may have been written by a "subject specialist" and may be very small and specific. Each Problem is comprised of many *Screens* and a *Logic* component. Each Screen contains a part of a Problem which the student views and can contain for example HTML, images, text gap (allows a text string to be inputted) and/or Multiple Choice Questions (MCQs). The Logic component specifies the movement between the Screens and is achieved by having several actions such as *Match, Jump* and *Terminate*. The Match action checks whether the student's answer to a question is correct; and the Jump action, depending on the outcome of the answer, specifies the Screen to jump to; and the Terminate action stops the Problem. A deterministic sequence of interactions is provided by the Logic so that two students, who provide identical responses to questions, will be presented with identical sequences of Screens.

A Roadmap is the logic mechanism which links together individual Problems to form a package of learning material (course) and/or assessment questions (test). Specific Problems to be presented are selected by the Roadmap based on the student's previous responses to Problems, i.e. the student's performance which is stored in a database. This means that each student is presented with a customised course and that the same Problems can be shared by different Roadmaps to form different courses hence is a more cost-effective way of developing and reusing learning resources (Mohan *et al*., 2003). The position within a particular Problem of each Roadmap can also be saved which allows students to return to it at a later point. Our Java course contains learning material which can be represented as four individual Roadmaps: *Pre-Test, Lesson Instructions, Performance Test* and *Proficiency Test* and the student's performance in each Roadmap is stored in the database.

Pre-Test

The Pre-Test Roadmap can be taken by students to establish their proficiency levels in each of the topics. Gaps in students' knowledge can be identified and the students are helped realise their weaker areas so that more time can be spent on these areas in the learning stage (Arroyo *et al*., 2001). The test begins with the easiest topic (*assigned with difficulty level 1*) and gradually increases difficulty level, topic by topic, until it reaches the most difficult topic (*assigned with difficulty level 10*). Figure 2 shows the simplified version of the Pre-Test within each topic. The horizontal arrows show that if a question is answered correctly, the next question is presented, which is of the same or higher proficiency level. The vertical arrows show that if a question has been answered incorrectly, the test is terminated and the student's proficiency level is established. Since this Pre-Test system is targeted at novice students, a more proficient student is nevertheless required to attempt all 6 questions.

Figure 2 The Simplified Pre-Test Model.



Lesson Instructions

Factual information, examples or code examples, pictures, tips for writing code of each topic are presented to students in this learning stage. Adamchik & Gunawardena (2003) argue that programming knowledge is best transferred from the teacher to the student by utilising LOs to present key programming concepts. These, constructed from a repository of information objects consisting of textbook content, code examples and review questions, can form a library of reusable LOs allowing different courses to be developed. The requirements of different students are met by establishing the learning material into different levels of proficiency and also by allowing an option to view more examples for those students who require more learning material (Wu, 2002). The results of the Pre-Test determine which level of proficiency of material is to be presented to the student. Exercises have also been integrated because they are an essential part of reinforcing a student's learning process. Answer explanations are presented before students proceed to the next Problem. The Roadmap is also able to select a more difficult question for a student who seems to be more proficient because he/she has not required viewing any of the extra examples.

Performance Test

This type of assessment can be regarded as a summative final exam which tests whether a student is able to complete a specific task based on the skills acquired during the learning stage. A final exam to create a Noughts and Crosses game was constructed for our Java course which examines the ten topics and this will be imported into OCTA. An overall score is allocated to the test; however, each question from the various topics is assigned a different weighted score according to its overall level of difficulty and its level of proficiency within each topic. This allows students of different levels of proficiency to be distinguished (Wainer *et al.*, 2000). A hint option is also available and if selected decreases the marks allocation for that question but will guide the student to the correct answer. Less proficient students will benefit from this as they may require help in arriving at the correct answer. Since the objective of this test is to help students realise the applications of the programming concepts previously learnt, the students may refer to the Lesson Instructions at any point necessary. Once all the questions have been completed, an overall score with a breakdown of marks gained from each topic is displayed. The student is recommended to return to the learning material of the weaker areas, if any.

Proficiency Test

This test is aimed at students who have previous Java programming experience and may not wish to participate in the learning process but only to establish their level of proficiency. The test begins at intermediate level because the learner's initial proficiency level is not available and therefore an intermediate level is assumed (Wainer *et al.*, 2000). Depending on whether the student answers correctly or incorrectly, the next question will be at advanced or novice level respectively. When a sufficient number of questions of the same proficiency level have been answered correctly, the test is terminated and the student's level of proficiency is established. Students find this type of assessment more interesting because the test is able to establish the student's proficiency level without having to ask numerous questions which may be too easy or too difficult for them (Gouli *et al.,* 2001).

Reusable Learning Objects

The learning needs of each student can be addressed by the LO approach and is the first step toward achieving personalized content (Adamchik & Gunawardena, 2003). We have adopted the reusable LOs approach in teaching students programming, since our course in OCTA can be regarded as composed of LOs and each LO can be reused in the different Problems and Roadmaps. A LO library containing many assessment questions allow them to be selected, used and reused in the many different tests. This improves the security of each test; this means that a student cannot perform well in a test by learning minimal material (Wainer *et al*., 2000).

There are a variety of ways which OCTA is able to import or export LOs to other learning environments, with minimal changes required. The recently approved *Learning Object Metadata* (LOM) *Standard* (IEEE, 2002) identifies many characteristics attributable to LOs and contains 76 data elements. These elements fall into nine categories: *General, Lifecycle, Meta-metadata, Technical,* *Education, Rights, Relation, Annotation* and *Classification*. A customised and simplified version of LOM will be constructed for OCTA which will have a selection of tags from the following categories: General, Technical, Education, Relation and Classification. The most relevant category for OCTA will be the Educational category which contains for example, description, learning resource type, intended end user role, typical learning time and age range. An extra category called *Assessment* will be added which contains for example, assessment topic and its difficulty level, specific material within the topic being tested and answer.

Related work includes Wu (2002) who defined a *Learning Object Mark-Up Language* which incorporated metadata for LOs, exercises and tests. Cesarini *et al.* (2004) has also encapsulated reusable tests in their LO framework. However, an alternative way is to use the *IMS Question and Test Interoperability Specification* (IMS, 2002) for setting and sharing web-based assessments (Lee *et al.*, 2004).

Conclusion

We have imported and transformed a non-adaptive course into an adaptive one in our computer-aided learning software OCTA. Pre-tests, lesson instructions, performance tests and proficiency tests have been identified as four areas of adaptive learning and testing. A learning object library has been formed which allows the reuse of learning objects in each of these different areas. We are merging the OCTA metadata which describes the learning objects with a simplified and customised version of *Learning Object Metadata* which will allow the import and export of LOs between learning environments. The software has been evaluated by a sample of students from the target audience with a variety of previous programming skills, and positive feedback was received from all of them.

References

Adamchik, V. & Gunawardena, A. (2003) A Learning Objects Approach to Teaching Programming. *International Conference on Information Technology: Computers and Communications 2003*, (pp.96-99).

Arroyo, I., Conejo, R., Guzman, E. & Woolf, B. P. (2001) An Adaptive Web-based Component for Cognitive Ability Estimation. University of Massachusetts, 2001.

ASTD & Smartforce (2002) A Field Guide to Learning Objects. Retrieved from: www.learningcircuits.org/2002/jul2002/smartforce.pdf (Accessed: 07/04/04)

Cesarini, M., Mazzoni, P. & Monga, M. (2004) Learning Objects and Tests. *The IASTED International Conference on Web-Based Education 2004*, (pp. 520-524)

Gouli, E., Kornilakis, H., Papanikolaou, K. & Grigoriadou, M. (2001) Adaptive Assessment Improving Interaction in an Educational Hypermedia System. *Conference on Human Computer Interaction 2001.*

Hockemeyer, C. & Dietrich, A. (2002) Adaptive eLearning and the Learning Grid. *LEGE-WG International Workshop on Educational Models for GRID Based Services.*

IEEE (2002) Draft Standard for Learning Object Metadata. Retrieved from: ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf (Accessed: 07/04/04)

IMS (2002) Question and Test Interoperability: An Overview (Final Specification Version 1.2). Retrieved from: www.imsglobal.org/question/qtiv1p2/imsqti_oviewv1p2.html (Accessed: 07/04/04)

Joy, M., Muzykantskii, B., Rawles, S. & Evans, M. (2002) An Infrastructure for Web-based Computer-Assisted Learning. *ACM Journal on Educational Resources in Computing*, 2(4), (pp. 1-19).

Lee, W. P. & Goh, A. (2004) Setting and Sharing Web-Based Assessments. *The IASTED International Conference on Web-Based Education 2004*, (pp. 269-274).

Mohan, P. & Greer, J. (2003) Reusable Learning Objects: Current Status and Future Directions. *World Conference on Educational Multimedia, Hypermedia and Telecommunications Volume 2003, Issue 1*.

W3C (2003) Extensible Markup Language (XML). Retrieved from: www.w3.org/XML/ (Accessed: 07/04/04)

Wainer, H. *et al.* (2000) *Computerized Adaptive Testing: A Primer* (Second Edition) Lawrence Erlbaum Associates Inc.: New Jersey.

Wu, H. (2002) Designing a Reusable and Adaptive E-Learning System. Masters Thesis, University of Saskatchewan. Retrieved from: www.cs.usask.ca/faculty/cooke/HGthesis.pdf (Accessed: 07/04/04)

Yau, J. (2002) Learning Java by Jane Yau: An Introductory Java Programming Course. Retrieved from: uk.geocities.com/jane_yau/ (Accessed: 07/04/04)