DESIGN, DEVELOPMENT, AND VALIDATION OF LEARNING OBJECTS

GWEN NUGENT

LEEN-KIAT SOH

ASHOK SAMAL

University of Nebraska-Lincoln

ABSTRACT

A learning object is a small, stand-alone, mediated content resource that can be reused in multiple instructional contexts. In this article, we describe our approach to design, develop, and validate Shareable Content Object Reference Model (SCORM) compliant learning objects for undergraduate computer science education. We discuss the advantages of a learning object approach, including positive student response and achievement, extension to other settings and populations, and benefits to the instructor and developers. Results confirm our belief that the use of modular, Web-based learning objects can be used successfully for independent learning and are a viable option for distance delivery of course components.

INTRODUCTION

Learning objects have their background in object-oriented paradigm of computer science focusing on the development of computer code components (called objects) that can be reused in multiple programming contexts. From an instructional standpoint, learning objects are small, stand-alone, mediated, content "chunks" that can be reused in multiple instructional contents, serving as building blocks to develop lessons, modules, or courses.

The value of learning objects has been touted by the Department of Defense [1], business and industry [2], public schools [3, 4], and higher education [5-8],

271

all citing: reusability; ease of updates, searches, and content management; customization, interoperability; and overall flexibility. Research on learning object approaches has also verified their instructional value [6, 9, 10]. Despite the current "hype" surrounding learning objects and their ostensible promise, however, there are potential drawbacks, including issues about willingness to share and reuse, the need for instructional context, the complications of developing complex metadata and incorporating SCORM standards, and the lack of formal design approaches [11-13]. This article discusses our experiences in designing, developing, and validating SCORM-compliant learning objects and presents both the benefits, as well as the challenges.

DESCRIPTION OF THE LEARNING OBJECTS

Our learning objects focused on the computer science topics of *simple class* and *recursion*. A class is a generalized or abstract definition of an object from which duplicate and/or modified versions may be generated and is a key concept in object-oriented programming. We use the term *simple class* to refer to the basic ideas behind the class concept. *Recursion* is a problem-solving technique that allows a problem to be solved using smaller instances of the same problem. Both these topics are recommended for inclusion in introductory computer science courses by the Association for Computing Machinery (ACM) and IEEE Computer Society, the two leading professional bodies in computer science [14]. The learning objects were designed in conformance with Shareable Content Object Reference Model (SCORM), a set of technical specifications originating from the Department of Defense and operationalized by the Advanced Distributed Learning (ADL) initiative [15]. Each learning object consisted of a series of Shareable Content Objects (SCOs), smaller content "chunks," representing specific examples, exercises, etc. which were assembled into a comprehensive lesson. Dividing each learning object into smaller segments was consistent with the SCORM premise that the reuse of content is dependent on the granularity of the content. By themselves, individual SCOs might not be able to teach a particular topic, but they were small enough to promote reusability and could be combined to provide a comprehensive instructional experience. Thus, the design process consisted of both carefully segmenting the content into discrete, coherent units, and also developing an overall sequencing scheme to provide the context necessary for student understanding.

Each learning object, with a glossary providing definitions to key terms and a help menu, consisted of four basic components:

• A brief tutorial or explanation provided definitions, rules, and principles. This portion consisted of background information and concepts and links to get additional information.

• A set of real-world examples illustrated key concepts and included worked examples and problems, models, and sample code. For example, one learning object used animation supported with accompanying narration to describe a CD player as a "class" with its data members and methods. (See Figure 1.)
• A set of practice exercises provided important active experiences to the student. When the learner made a mistake, the learning object not only noted the error, but also provided a detailed explanation and gave the correct answer. Figure 2 shows a screen shot of one of the practice exercises focusing on data members and methods from the *simple class* learning object.
• A set of problems graded by the computer provided a final self-test assessment. The assessment was presented only after a learner had scored sufficiently high in the practice exercises.

The learning objects made extensive use of Flash animation and utilized multiple user input formats, including drag-and-drop, multiple choice, and model
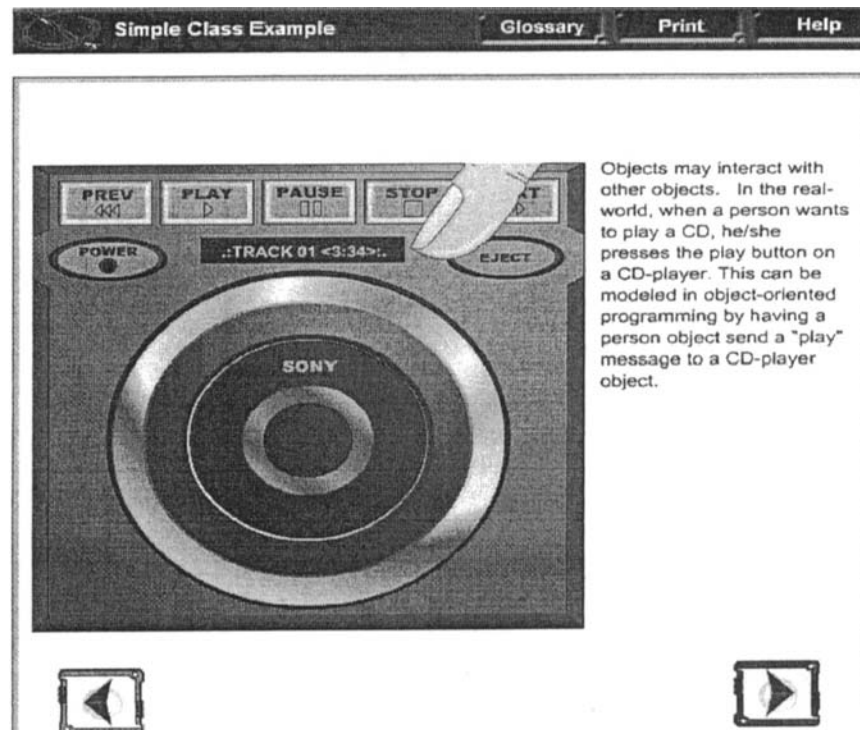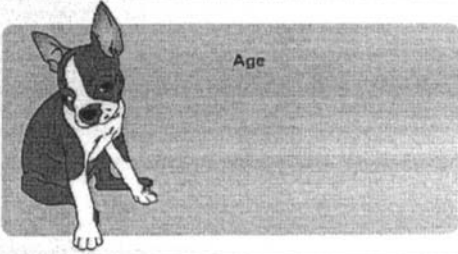

Figure 1. Screen shot of CD example.

Figure 2. Screen shot of practice exercises component
showing feedback.

construction. They were designed to allow students to move ahead quickly if they had a good understanding of the concepts. Students who provided correct responses had the choice of working on additional problems or progressing to the next activity. For students requiring additional instruction, appropriate feedback was provided. The feedback function was critical because it contained significant instructional content.

The system also included customized tracking functions that automated the data gathering process. The data collection process was designed to track time spent on individual activities, paths taken, and choices made. This tracking capability provided valuable feedback both to the instructor and multimedia developers. Data could be displayed for individual students or aggregated in various groupings depending on the demographic variable of interest. Although we originally used a tracking system developed in-house, we are now exploring the data collection capabilities available through SCORM and the learning management system (LMS).

DESIGN AND DEVELOPMENT

Although learning objects have been in use for more than a decade, much of the effort has focused on technology and standards, and few formal design approaches have been developed [11-13]. There is little concern for instructional design practice within the technical specifications of SCORM. The one exception is that SCORM version 1.3 introduced standards on sequencing, a key issue in our instructional design approach. Our approach also relied on traditional principles from cognitive theories of multimedia learning [16] and cognitive load [17]. Multimedia learning theory provided guidance for the effective combination of text, graphics, audio, and Flash animation. For our particular audience, novice learners encountering complex content, principles of cognitive load theory were also important. Our design of learning objects focused on appropriate use of multimedia elements, student practice, feedback, and guidance, with the goal of encouraging students to be cognitively active while minimizing cognitive load demands.

Each of the four learning object components also had a basis in theory and/or research. For example, the opening tutorial provided background information needed by the learner to activate the prior knowledge necessary to learn new concepts. Concrete, authentic examples and problems were selected based on research showing their importance for improving student learning and motivation for complex subject matter [18]. The use of worked examples has shown distinct benefits for reducing extraneous cognitive load and promoting schema formation [19, 20]. Immediate elaborative feedback also guides students in the process of understanding computer science concepts [21-23].

RESEARCH DESIGN

We used both evaluation and research approaches in testing the learning objects. The *simple class* learning object was first piloted with computer science students in Spring 2004. Students were given two weeks to review and evaluate the multimedia material. They completed a survey with nine Likert scale questions assessing various dimensions of the mediated materials (Table 1). There were also open-ended questions soliciting specific comments about each of the sections and the learning object as a whole.

This evaluation was followed the subsequent semester by more refined research testing the effectiveness of the Web-based learning objects by comparing the learning of students who participated in a traditional computer science laboratory (completing in-class exercises with guidance from the instructor) with students who completed the Web-based learning objects. Students using the learning objects did so in the regular laboratory classroom. Both groups took a 10-item posttest, with multiple choice questions at a variety of Bloom's Taxonomy levels [24], developed for the laboratory sessions on *simple class* and *recursion*. These

Table 1. Likert-Type Survey Questions to Evaluate the
Simple Class Learning Object (*n* = 33)

No.	Question	Mean	SD
1	The simple class learning module was easy to use.	4.56	.50
2	The learning object maintained my interest.	4.25	.56
3	I learned a lot from the learning object.	3.72	.84
4	The graphics added a lot to the content presentation.	4.38	.65
5	The learning object is a valuable addition to the course.	4.47	.61
6	More of the course material should be presented through the Web.	4.03	.88
7	The learning object helped me understand more about simple class.	3.97	.88
8	I will use the learning object again in the future if I have questions about simple class.	3.53	.97
9	Overall, how would you rate the learning object on simple class? (Poor, fair, so-so, good, excellent)	4.31	.53

1 = strongly disagree, 2 = disagree, 3 = unsure/neutral, 4 = agree, 5 = strongly agree.

posttests had been previously prepared as part of a department initiative to add a laboratory component to support the lecture portion of the course. Posttests for each of the laboratory sessions were a major component of the students' final grade.

Random assignment into the two treatment groups was made by lab section. One lab section participated in the traditional laboratory activities, the other spent the lab time completing the Web-based learning object. Because the treatment conditions (learning object vs. traditional lab) were not randomly assigned to individual students, equivalence of student computer science knowledge and abilities between lab sections was critically important. To test for group equivalency, we examined student scores on the department's computer science pretest given at the beginning of the semester. There was no significant difference between mean pretest scores for the two lab sections (traditional lab $M = 26.42$, learning object $M = 26.88$, $t(48) = .20$, $p = .84$). We also examined final homework and exam scores for the two groups (total possible homework points was 650 and total possible exam score was 300). Again, there was no significant

difference in the two groups' homework scores (traditional lab $M = 466.28$, learning object $M = 449.60$, $t(35) = .35$, $p = .73$) and exam scores (traditional lab $M = 228.05$, learning object $M = 205.11$, $t(35) = 1.63$, $p = .11$).

The simple class and recursion posttest scores data were analyzed using two independent measure t-tests. One test determined any differences between traditional laboratory activities and learning object for the *simple class* topic; the other tested for differences between the same experimental conditions for *recursion*.

RESULTS

Student self-report evaluation results are presented and discussed first, followed by the research results focusing on student achievement.

Evaluation Results

Results are discussed in terms of descriptive statistics (Table 1) and qualitative, open-ended responses. Results showed that students found the *simple class* learning object easy to use and a valuable addition to the course. It maintained their interest and helped them better understand the topic. Students felt the graphics were important to the content presentation. Overall, they rated the learning object to be better than "Good-Excellent." They generally approved of the design (questions 1, 2, and 4), appropriateness (questions 5 and 9), and usefulness (questions 3, 6, 7, and 8). Open-ended comments from the students also provided evidence of their effectiveness: "I really like this program and what it taught me"; "I like the real-time feedback"; "It is very informative, interactive, and fun"; "I liked how you could click glossary, print, or help on every page."

Research Results

Comparisons between student learning from the traditional laboratory activities versus the learning objects showed no significant differences for both the simple class and recursion topics (Table 2). Results show the approximate equivalence of the learning object and the traditional laboratory experience in promoting student learning.

Table 2. Comparison of Mean Achievement Test Scores for Traditional Versus Learning Object Approach

Topic	Traditional lab	Learning object	t-Test value	p Value
Simple Class	8.29	7.88	$t(48) = 1.04$	.30
Recursion	9.39	8.84	$t(35) = 1.41$	.17

CHALLENGES

The benefits derived from the learning object approach were not without significant challenges—primarily related to the implementation of SCORM standards. Our multimedia producers had extensive experience in implementing SCORM standards in a courseware production environment [25]. However, standards changes between SCORM versions caused major problems. In the middle of the development process, our University upgraded its Blackboard course management system and our learning objects, designed to conform to SCORM version 1.2, had problems executing within the new learning management system (LMS) environment. Significant time and expense were necessary to upgrade the material to work with SCORM 1.3, which the new version of Blackboard supported.

Other challenges included the time commitment and extensive interaction needed between the subject matter experts, the instructional designers, and the multimedia producers. The computer science faculty, unaccustomed to working with a multimedia development team, were surprised by the length of time for the development process. They found many aspects of the script development process repetitive and boring and the need for constant review and debugging to be time consuming. The team is now exploring the development of templates which can streamline the process for content development, scripting, and student tracking and can be used by upper-class undergraduate computer science students to produce the Web-based instruction.

CONCLUSIONS

We have described our approach to design, develop, and validate learning objects for undergraduate computer science. Our evaluation of the *simple class* learning object yielded encouraging results, and provided good comments to guide revisions and guidance for the development of the *recursion* learning object. The data obtained from the tracking software was also helpful in determining areas where revisions were needed. Our more refined experimental approach showed equal effectiveness of the learning objects and the traditional laboratory activities. The series of evaluation and research results confirm our belief that the use of modular, Web-based learning objects can be used successfully for independent learning for complex subject matter and are a viable option for distance delivery of course components. This is important because the learning objects have many non-learning benefits. For example, their automatic grading provides efficiency benefits for the instructor. The software tracking features, such as time on task and learning progress, provide valuable feedback to the instructor and multimedia developer.

We are especially pleased that high school instructors and administrators have requested access to our learning objects. Because the learning objects have been designed as stand-alone instruction, they can be used to support high school classroom instruction, as well as for self-study opportunities for rural students interested in computer science. Providing our learning objects to the high schools, however, required moving the content from Blackboard, used in our university environment, to Angel, the LMS used by our state's public school distance learning consortium. While we were concerned that the transfer process might present problems, we were pleased that the learning objects played perfectly in the Angel LMS—an example of achieving the SCORM goal for the reuse of content among many learning management systems.

Future activities include analyzing results based on student pretest scores and final grade to determine the impact of the learning objects with different groups of learners. Evaluation results and open-ended student comments suggest that the learning objects differentially benefit students having difficulty and that learning objects are a valuable way to present topics that students perceive as more difficult. Our long-term plan is to identify, design, and build a suite of learning objects for introductory computer science curriculum and extend them to high schools to better prepare students for college level computer science classes. We are also exploring the development of tools to make the design and development of learning objects more efficient.

REFERENCES

1. Advanced Distributed Learning, *DoD Affirms SCORM's Role in Training Transformation*, ADL News Release, December 15, 2003. Retrieved July 18, 2005 from: www.adlnet.org
2. ASTD, A Primer on Learning Objects, *Learning Circuits: ASTD's Online Magazine about E-learning*, 2000. Retrieved October 10, 2004 from: http://www.astd.org/ASTD/Resources/dyor/article_archives.htm
3. P. Grunwald, *Video and Television Use Among K-12 Teachers*, Power Point prepared for PBS, 2002.
4. S. Pasnik and D. Keisch, *Teachers' Domain Evaluation Report*, Center for Children and Technology, New York, 2003. Retrieved October 10, 2004 from: http://www2.edc.org/CCT/publications_report_summary.asp?numPubId=148
5. T. Koppi and N. Lavitt, *Institutional Use of Learning Objects Three Years On: Lessons Learned and Future Direction*, 2003. Retrieved October 10, 2004 from: http://www.cs.kuleuven.ac.be/~erikd/PRES/2003/LO2003/
6. C. Bradley and T. Boyle, *The Development and Deployment of Multimedia Learning Objects*, 2003. Retrieved October 10, 2004 from: http://www.cs.kuleuven.ac.be/~erikd/PRES/2003/LO2003/

7. G. Francia, A Tale of Two Learning Objects, *Journal of Educational Technology Systems*, *3*:2, pp. 177-190, 2003.

8. E. Van Zele, P. Vandaele, D. Botteldooren, and J. Lenaerts, Implementation and Evaluation of a Course Concept Based on Reusable Learning Objects, *Journal of Educational Computing Research*, *28*:4, pp. 355-372, 2003.

9. F. J. Boster, G. S. Meyer, A. J. Roberto, and C. C. Inge, *A Report on the Effect of the United Streaming Application on Educational Performance*, Cometrika/United Learning, 2002.

10. A. Samal, G. Nugent, L. K. Soh, and J. Lang, Reinventing Computer Science Curriculum at the University of Nebraska, in *Technology-based Education: Bringing Researchers and Practitioners Together*, L. Pytilk-Zillig, M. Bodvarsson, & R. Bruning (eds.), Information Age Publishing, Greenwich, Connecticut, pp. 63-82, 2005.

11. P. Parrish, The Trouble with Learning Objects, *Educational Technology Research and Development*, *52*:1, pp. 49-67, 2004.

12. D. Wiley, *Learning Objects: Difficulties and Opportunities*, 2000. Retrieved July 18, 2005 from: http://wiley.ed.usu.edu/docs/lo_do/pdf

13. E. D. Wagner, The New Frontier of Learning Object Design, *The E-Learning Developer's Journal*, June 2002.

14. G. Engel and R. Eric (Eds.), *Computing Curricula 2001: Computer Science Volume*, ACM Press, New York. Retrieved May 12, 2004 from: http://www.sigcse.org/cc2001/

15. Advanced Distributed Learning, *Shareable Content Object Reference Model*, 2001. Retrieved May 23, 2005 from: http://www.adlnet.org/scorm/index.cf

16. R. E. Mayer, *Multimedia Learning*, Cambridge University Press, New York, 2001.

17. J. Sweller, *Instructional Design in Technical Areas*, ACER Press, Camberwell, Australia, 1999.

18. J. Van Merrienboer, R. Clark, and M. De Croock, Blueprints for Complex Learning: The 4C/ID-Model, *Educational Technology Research and Development*, *50*, pp. 39-64, 2002.

19. F. Pass and J. J. G. Van Merrienboer, Variability of Worked Examples and Transfer of Geometrical Problem-Solving Skills: A Cognitive Load Approach, *Journal of Educational Psychology*, *86*, pp. 122-133, 1994.

20. J. Sweller, J. J. G. Van Merrienboer, and F. Pass, Cognitive Architecture and Instructional Design, *Educational Psychology Review*, *10*, pp. 251-296, 1998.

21. J. E. Brophy and T. L. Good, Teacher Behavior and Student Achievement, in *Handbook of Research on Teaching* (3rd Edition), M. C. Wittrock (ed.), Macmillan, New York, pp. 328-375, 1986.

22. R. Moreno, Decreasing Cognitive Load for Novice Students: Effects of Explanatory Versus Corrective Feedback in Discovery-Based Multimedia, *Instructional Science*, *32*, pp. 99-113, 2004.

23. B. Rosenshine and R. Stevens, Teaching Functions, in *Handbook of Research on Teaching* (3rd Edition), M. C. Wittrock (ed.), Macmillan, New York, pp. 376-391, 1986.

24. B. S. Bloom, *Taxonomy of Educational Objectives: Book 1, Cognitive Domain*, Longman, New York, 1956.

25. B. Barker, Adopting SCORM 1.2 Standards in a Courseware Production Environment, *International Journal of E-Learning*, pp. 21-24, July-September, 2004.

Direct reprint requests to:

Dr. Gwen C. Nugent
Nebraska Center for Research on
   Children, Youth, Families, and Schools
University of Nebraska-Lincoln
216 Mabel Lee Hall
P.O. Box 880235
Lincoln, NE 68588-0235
e-mail: gnugent@unl.edu