2009

Empirical Usage Metadata in Learning Objects

Gwen Nugent  
*University of Nebraska - Lincoln*, gnugent@unl.edu

Kevin Kupzyk  
*University of Nebraska - Lincoln*, kkupzyk@unl.edu

S. A. Riley  
*University of Nebraska - Lincoln*, sriley@cse.unl.edu

L.D. Miller  
*University of Nebraska*, lmille@cse.unl.edu

Jesse Hostetler  
*University of Nebraska - Lincoln*, jhostetl@cse.unl.edu

*See next page for additional authors*

Follow this and additional works at: [http://digitalcommons.unl.edu/cseconfwork](http://digitalcommons.unl.edu/cseconfwork)

Part of the [Computer Sciences Commons](http://digitalcommons.unl.edu/computer-sciences-commons)


[http://digitalcommons.unl.edu/cseconfwork/19](http://digitalcommons.unl.edu/cseconfwork/19)

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in *CSE Conference and Workshop Papers* by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.
Authors

Gwen Nugent, Kevin Kupzyk, S. A. Riley, L.D. Miller, Jesse Hostetler, Leen-Kiat Soh, and Ashok Samal
Empirical Usage Metadata in Learning Objects

Gwen Nugent, Kevin Kupzyk, S. A. Riley, L. D. Miller, Jesse Hostetler, Leen-Kiat Soh, Ashok Samal
University of Nebraska-Lincoln, gnugent@unl.edu, kkupzyk2@unl.edu, sriley@cse.unl.edu, lmille@cse.unl.edu, jhostel@cse.unl.edu, lksoh@cse.unl.edu, samal@cse.unl.edu

Abstract - The iLOG Project (Intelligent Learning Object Guide) is designed to augment multimedia learning objects with information about (1) how a learning object has been used, (2) how it has impacted instruction and learning, and (3) how it should be used. The goal of the project is to generate metadata tags from data collected while students interact with learning objects; these metadata tags can then be used to help teachers identify learning objects that match the educational and experiential backgrounds of their students. The project involves the development of an agent-based intelligent system for tracking student interaction with learning objects, in tandem with an extensive learning research agenda. This paper provides an overview of this NSF-funded project, focusing on the instructional approach and research on varying levels of active learning and feedback. Using a randomized design and a hierarchical linear modeling framework, research showed that the active learning conditions resulted in significantly higher student learning. The elaborative feedback results approached (p = .056), but did not reach, the established significance criteria of alpha = .05. Both active learning conditions and one of the elaborative feedback conditions resulted in significantly higher content assessment scores compared to a control group.

Index Terms – Learning objects, active learning, feedback, computer science education

INTRODUCTION

From an instructional standpoint, learning objects (LOs) are stand-alone content that can be reused in multiple instructional contexts, serving as building blocks which can be assembled to create lessons, modules, or courses. Research on learning objects has verified their instructional value [1]-[3]; however, there are challenges in both their design and their search and retrieval. The iLOG project is an attempt to address both these challenges.

I. LO Design

Learning objects vary in their instructional effectiveness—inevitably, some LOs will, on average, “work” better than others. In order to create maximally effective LOs, courseware designers need concrete, empirically-based guidelines for LO design. Unfortunately, there is a lack of formal design approaches for LO development [4]-[6]. Therefore, our approach to the design of learning objects has relied on principles drawn from learning theory and research. Each of our LOs includes (1) a tutorial covering the topic, (2) a set of ungraded exercises, and (3) a set of graded assessment questions. The opening tutorial provides background information needed by the learner to activate prior knowledge necessary to learn new concepts [7]. The exercises provide student practice with the concepts [8], and the assessment is intended to provide information to the student as to the level of his/her understanding [9]. We have also relied on cognitive theories of multimedia learning [10] which has provided guidance for effective combination of text, graphics, audio, and Flash animation. Our design of learning objects has focused on appropriate use of these multimedia elements, student practice, feedback and guidance, with the goal of encouraging students to be cognitively active. Drawing on this theoretical and research base, our learning research agenda is focused on utilizing the most effective approaches for learning object instruction. The research described in this paper focuses specifically on the use of active versus passive learning and the use of elaborative feedback versus simple knowledge of results. Both elements have critical implications for student learning, as well as learning object design and production. There is an increased production cost associated with having extensive opportunities for student interactions during content presentation. Similarly, the development of elaborative feedback, which targets correcting student errors and misconceptions, is extremely time consuming to prepare and requires thorough understanding of how students learn computer science concepts.

II. LO Search and Retrieval

In order to fully exploit the modularity and reusability of well-designed LOs, there must be a means available to describe them, organize them into repositories, and search within these repositories for LOs matching specific requirements. However, the process of accurately describing an LO in a way that facilitates automated searching is not straightforward or readily automated. One common solution is to tag each LO with descriptive metadata. This has traditionally been a manual process carried out by courseware authors, but the end result has often been LO metadata that is subjective or inaccurate at best, and at worst entirely missing. Moreover, a recent study [11], suggests that the most common LO metadata standard, the IEEE 1484.12.1 – 2002 Standard for Learning Object Metadata (LOM), is currently insufficient for the automatic classification and discovery of LOs. It should be noted that automatic extraction of metadata from an LO is an active
area of research; however, even the most recent methods [12] have only been able to extract partial metadata records from the learning content.

The discovery process is further complicated by variability in learning characteristics of the students such as background knowledge, experience, motivation, and self-efficacy, which have a strong influence the learning outcomes for a particular student interacting with a particular LO. Thus, it is not enough to simply locate an LO on the right topic—we would also like to locate an LO that is appropriate to an individual student's learning context. Research on the detection of student characteristics such as motivation and self-efficacy from interaction log file data is also an active area of research [13]; however, to our knowledge, the automatic generation of searchable, contextual LO metadata from usage log files is a novel approach.

Thus, tagging an LO with metadata that provides a meaningful description of how it has been used and how it should be used is inherently uncertain. This, in turn, makes the search for and selection of an appropriate LO a nontrivial process; it may even be daunting enough to send an instructor back to the chalkboard and student back to the textbook.

The iLOG Project (Intelligent Learning Object Guide) aims to help solve these issues of learning object design and retrieval. To address the challenges and opportunities in LO design, we revised and converted existing course materials for undergraduate introductory computer science courses into LOs; at the same time we implemented an active learning research agenda focusing on key instructional design strategies and identification of the student characteristics that are most useful in understanding and diagnosing student success and failure. To assist in the intelligent search and retrieval of LOs, we created a system that tracks, diagnoses and tags LOs. In this paper, we briefly introduce the iLOG system and discuss the results of our experiments, which compared the effectiveness of several versions of each LO; each designed with varying levels of active learning and feedback.

Session W1F

In the following, Section 2 describes the technologies used in the iLOG project to facilitate the automated collection of empirical usage data and the subsequent analysis of such data to generate metadata. Section 3 details our studies involving active learning and elaborative feedback in our LO design. Sections 4 and 5 present and discuss the results of our semester-long study covering four different courses. Finally, we conclude and outline our ongoing work and future work.

THE ILOG PROJECT: TECHNOLOGY

A main technology goal of the iLOG (Intelligent Learning Object Guide) Project is to create a system to track, diagnose, and generate empirical usage metadata for Sharable Content Object Reference Model (SCORM) conformant LOs. Figure 1 illustrates the two main components in the iLOG system: (1) the LO wrapper, which is responsible for logging student interactions and updating the LO metadata, and (2) MetaGen, which processes the data gathered by the wrapper and generates empirical usage metadata.

The LO wrapper is an HTML document based on the Easy SCO Adapter [http://www.ostyn.com]. The wrapper encapsulates the LO, forming a transparent layer between the LO and its host learning management system (LMS). When the LMS and the LO communicate using the standard SCORM API functions (e.g. when the student submits a response to a graded question), or when certain HTML events are generated by the LO (e.g. when the student navigates between pages), the wrapper uses Javascript functions to monitor and collect this interaction data. When the LO session terminates, the wrapper then uploads the data to the MetaGen component, located on a remote server.

The MetaGen component includes three modules: (1) a data logging module, (2) a data extraction module, and (3) a data analysis module. The data logging module uses the PHP scripting language to receive student interaction data from the LO wrapper and store it in a MySQL database. Once these data have been collected, the data extraction module queries the database and processes the data to obtain the iLOG dataset.

Figure 1

COMPONENTS FOR THE INTELLIGENT LEARNING OBJECT GUIDE.
The data analysis module is composed of several complementary steps: generation of usage statistics, feature selection, and data mining. First, the data analysis module uses the Weka [14] implementations of several feature selection algorithms to choose the subset of data features that are most salient to the learning process. This step may provide useful insights in and of itself, because it highlights the student characteristics and patterns of interaction behavior that are correlated with different learning outcomes. Next, the data analysis module runs the Tertius [15] predictive rule mining algorithm on the selected feature subset to generate empirical usage metadata. In addition to the generation of these prediction rules, the data analysis module also calculates empirical usage statistics such as average time spent on the LO and average assessment scores.

The current study focuses on the evaluation of the impact of varying the levels of active learning and feedback in an LO. The research uses the dataset generated by the data extraction module; the dataset was imported into a statistical processing package for analysis. For a more detailed description of the use of the complete iLOG framework for automatic generation of empirical usage metadata we refer the reader to [16].

THE ILOG PROJECT: ACTIVE LEARNING AND ELABORATIVE FEEDBACK RESEARCH

A primary learning goal of the iLOG project is to measure the impact of active learning and elaborative feedback on student learning. E-learning, through the use of LOs, provides an excellent opportunity to prompt active student response and provide ongoing feedback that is targeted to address individual student errors.

Learning theory and research emphasize that learning is enhanced by actively engaging students in the learning process [17], [18]. In contrast to passively listening to a lecture or viewing a demonstration, active learning requires students to make decisions and choices. For our experiment to test the use of active learning, we developed two versions of each LO for comparison. In the active learning versions, students completed exercises that required them to manipulate graphical objects on the screen. In the corresponding passive learning versions a predetermined sequence of responses to the same exercises was demonstrated to the students in the form of a video.

Feedback is another critical instructional design dimension. Studies of learning, transfer, and development show that feedback is extremely important and that, usually, it should be immediate [2] [19]-[22]. A review [23] of 3000 research reports showed a feedback effect size of .4, which could be considered a medium sized treatment effect. Another study [24] showed that as the level of feedback increased, so did the effect size. Our feedback research focused on two levels: (1) low level feedback, as reflected by simple knowledge of results, and (2) elaborative feedback, as reflected in extensive explanations and models.

METHODOLOGY

I. Participants

The participants were 189 undergraduate students from the University of Nebraska-Lincoln enrolled in introductory computer science courses. These students completed the LOs as part of their graded course work and represented a wide variety of backgrounds and learning contexts (e.g. non-majors, majors, and honors students).

II. Description of Instructional Conditions and Research Design

To carry out this research, we created five SCORM-conformant LOs covering basic computer science concepts such as conditional statements, logic, arrays, and looping. Each includes (1) a tutorial covering the topic, (2) a set of ungraded exercises, and (3) a set of graded assessment questions. For each LO, four versions were developed: (1) active learning and elaborative feedback, (2) active – nonelaborative, (3) passive learning – elaborative, and (4) passive – nonelaborative. The four versions of instruction resulted in a 2 x 2 design: the Active vs. Passive factor and the Elaborative vs. Nonelaborative factor.

The research design also included a control condition, in which students still participated in traditional lecture and lab activities but were not exposed to the tutorial or the exercises in the LOs before taking the LO assessments. The control condition was assigned to assess improvements associated with the addition of the learning objects compared to the “business as usual” approach.

The LOs were deployed to students using the SCORM LMS from the Blackboard Academic Suite [http://www.blackboard.com/]. Students were randomly assigned to one of the five conditions (each of the four versions of instruction plus the control condition) for each LO. Because the students’ scores on the LOs affected their course grades, each student was assigned to each of the five conditions exactly once.

III. Data Analysis

In order to use all available data and account for the fact that students participated in up to five LOs, a hierarchical linear modeling (HLM) framework was used to analyze the data. This statistical framework estimates linear equations using nested data where outcomes for members of groups can be a function of the characteristics of the groups as well as the characteristics of the members. The clustering manifested in this study is LOs nested within students. An LO variable was then entered as a level 1 fixed effect. Recognizing the wide variety of student backgrounds, gender and student scores on the Computer Science Department’s placement test [25] (used to place students in the appropriate introductory computer science course) were used as covariates. The outcome measure (dependent variable) was student scores (percent correct) on the graded assessment portions of each of the five learning objects.
The effect of gender was found to be non-significant and thus was removed from the analytic model. The 2 x 2 ANOVA resulted in no interaction effect between active learning and feedback (F(1,394.4) = .509, p = .476), which allowed us to focus on main effects for these factors. This means that we can separately evaluate active learning and feedback. The scores on the LO assessments were significantly higher for students in the active versus passive learning condition (F(1,390.0) = 4.62, p = .032). Although scores for the elaborative feedback condition were higher than those for the nonelaborative condition, the difference was not significant (F(1,403.6) = 2.42, p = .121). The effects of learning object (F(4,414.2) = 16.16, p < .001) and placement test score (F(1,1187.3) = 75.21, p < .001) both accounted for a significant amount of variation in LO assessment scores.

Two models were estimated prior to this analysis, which excluded the effects of learning object topics and placement test scores, respectively. The significant difference between active and passive learning conditions was also significant in these models. If adding the effects of learning object or placement test scores had weakened the main effect of interest, the causal validity of the active condition effect may have been threatened. However, in the final analytic model the active condition effect remained unaffected by the addition of learning object topic and placement test scores as control variables, thus enhancing the causal validity of the main effect of active versus passive learning.

The follow-up analysis showed that, controlling for learning object and placement test scores, the active-elaborative, active-nonelaborative, and passive-elaborative versions of the learning objects resulted in significantly higher assessment scores than did the control condition. The passive-nonelaborative version did not result in significantly higher scores, the .055 significance level (one-tail test) is very close to the accepted alpha = .05. Of note is that the passive-nonelaborative condition, in which students were given no opportunity for active engagement or explanatory feedback, was no more effective than the control condition. Unfortunately, this is the instructional paradigm most often used in the college classroom. In such situations, the delivery of information is typically one-way from the instructor to the student, with little chance for students to respond in any sustained fashion, and with limited instructor feedback to address individual student questions and concerns.

Also important is the lack of an interaction between instructional condition and LO topic, showing that the effectiveness of each instructional strategy is consistent across the different content topics. Thus, the effectiveness of active and elaborative instructional strategies is not influenced by the nature and/or difficulty of the content. Nevertheless, the content area (as represented by the topics covered in the five LOs) and student scores on the Computer Science placement test are significant predictors of student learning, both accounting for a significant amount of variation in content test scores.

We expected our results along the feedback dimension to support existing learning theory and research and exhibit a significant effect on learning outcomes. One possible explanation for not observing this result may be that students are clicking past the feedback messages without reading them. In the case of the Active vs. Passive factor, it is impossible for students to elect to not participate in their assigned condition. For instance, a student cannot complete the interactive exercises without active engagement, and cannot interact with the video in the passive condition. However, students in both the minimal and elaborative feedback conditions can elect to ignore the feedback.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Observed Percent Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Elaborate</td>
<td>0.70</td>
</tr>
<tr>
<td>Active Nonelaborate</td>
<td>0.55</td>
</tr>
<tr>
<td>Control</td>
<td>0.30</td>
</tr>
<tr>
<td>Passive Elaborate</td>
<td>0.20</td>
</tr>
<tr>
<td>Passive Nonelaborate</td>
<td>0.10</td>
</tr>
</tbody>
</table>

In this paper we have described the technology and education goals of the cLOG project (Intelligent Learning Object Guide). The cLOG framework aims to automatically generate empirical usage metadata that describes actual student use of learning objects as a means to search and select learning objects. A key technology contribution of this work is the generation of metadata that facilitates the location of LOs that match a student’s learning context, which includes students’ prior knowledge, and subjective factors such as motivation to learn, and confidence in ability to learn the subject matter.

**Conclusions and Future Work**

In this paper we have described the technology and education goals of the iLOG project (Intelligent Learning Object Guide). The iLOG framework aims to automatically generate empirical usage metadata that describes actual student use of learning objects as a means to search and select learning objects. A key technology contribution of this work is the generation of metadata that facilitates the location of LOs that match a student’s learning context, which includes students’ prior knowledge, and subjective factors such as motivation to learn, and confidence in ability to learn the subject matter.
The focus of this paper was an experimental study related to a key education research goal of the iLOG project: to study the impact of varying levels of active learning and feedback on student outcomes when interacting with an LO. We designed 5 LOs covering a variety of topics in Introductory Computer Science and generated 5 different versions of each LO in order to conduct a study of the interactions of instructional strategies in a 2x2 experimental design with an Active vs. Passive factor, and Elaborative vs. Nonelaborative factor. A control group was used to assess the value of the learning object compared to the “business as usual” approach. These LOs were assigned to 189 introductory computer science students as part of their regular coursework.

One avenue for future work is to investigate the reason the elaborative feedback strategy did not show a significant impact on learning outcomes. A more thorough quantitative analysis of individual questions on the assessments may provide greater insight into the questions most often missed, and the student characteristics that were correlated with these mistakes. This information can guide generation of feedback that is more closely targeted to individual needs. This learning information, coupled with statistics on length of time students spend on individual questions or on exercises sections containing the feedback, can provide meaningful insight into the instructional design of effective learning objects. Furthermore, the use of the repository of individual and cumulated student usage data collected by the iLOG framework can provide valuable guidance to teachers and instructional designers seeking to improve computer science instruction.

ACKNOWLEDGMENT

This material is based on work supported by the National Science Foundation under Grant No. IIS-0632642, an NSF REU grant, two NSF GAANN fellowships, and a University of Nebraska’s UCARE grant.

REFERENCES


AUTHOR INFORMATION

Gwen Nugent, Research Associate Professor, Nebraska Center for Research on Children, Youth, Families and Schools, University of Nebraska-Lincoln, gnugent@unl.edu.

Kevin Kupzyk, Ph. D. candidate, Department of Educational Psychology, Quantitative and Qualitative Psychometric Methods, University of Nebraska-Lincoln, kkupzyk2@unl.edu.

Sarah Riley, Ph.D. candidate, Department of Computer Science and Engineering, University of Nebraska-Lincoln, sriley@cse.unl.edu

L.D. Miller, Ph.D. candidate, Department of Computer Science and Engineering, University of Nebraska-Lincoln, lmille@cse.unl.edu

Jesse Hostetler, B.S. candidate, Department of Computer Science and Engineering, University of Nebraska-Lincoln, jhostel@cse.unl.edu

Leen-Kiat Soh, Associate Professor, Department of Computer Science and Engineering, University of Nebraska-Lincoln, lksoh@cse.unl.edu

Ashok Samal, Professor, Department of Computer Science and Engineering, University of Nebraska-Lincoln, samal@cse.unl.edu