Evaluating the Use of Learning Objects in CS1

L.D. Miller, Leen-Kiat Soh, Ashok Samal
Department of Computer Science and Engineering
University of Nebraska
Lincoln, NE 68588
1-402-472-3200
{lmille, lksoh, samal}@cse.unl.edu

Gwen Nugent, Kevin Kupzyk,
Leyla Masmaliyeva
Center for Research on Youth, Family and Schools
University of Nebraska
Lincoln, NE 68588-0342
1-402-472-2448
{gnugent1, kkupzyk2}@unl.edu
leylamasmaliyeva@yahoo.com

ABSTRACT
Learning objects (LOs) have been previously used in computer science education. However, analyses in previous studies have been limited to surveys with limited numbers of LOs and students. The lack of copious quantitative data on how LOs impact student learning makes detailed analysis of LO usefulness problematic. Using an empirical approach, we have studied a suite of LOs, comprehensive in both the content covered and the range of difficulty, deployed to CS1 courses from 2007-2010. We review previous work on predictors of achievement and impact of active learning and feedback. We also provide a high-level overview of our LO deployment. Finally, based on our analysis of student interaction data, we found that (1) students using LOs have significantly higher assessment scores than the control group, (2) several student attributes are significant predictors of learning, (3) active learning has a significant effect on student assessment scores, and (4) feedback does not have a significant effect, but there are variables with significant moderating effects.

Categories and Subject Descriptors
K.3.2. [**Computer and Education**]: Computer and Information Science Education.

General Terms
Experimentation

Keywords
learning objects, self-efficacy, motivation, active learning, feedback

1. INTRODUCTION
Learning objects (LOs) have been used in computer science education as part of e-learning or supplements to traditional classroom lectures and labs. Their usefulness has been investigated but most studies employed pre- and post-treatment knowledge tests or surveys [17]. In-depth analyses have been scarce due to the limited nature of their deployment, both in number of LOs and the number of students involved [19]. Therefore, there is a need (1) to better understand how the use of LOs can impact student learning under different instructional and pedagogical situations, (2) to gain insights to facilitate e-learning in CS education in particular, and (3) to help address the heterogeneity of educational activities in general.

Our research takes an empirical approach to address these issues. The strength of our empirical approach is three-fold: First, we deployed a suite of 16 LOs on computer science concepts to CS1 courses involving more than 1100 students. This suite of LOs is comprehensive in terms of both the content covered and the range of difficulty. The large number of students minimizes variance on our analysis of active learning and feedback in the LOs. On the other hand, LOs on different topics allow more general predictors of student achievement. Second, we use a software tool to automatically collect and store all student interaction data, from mouse clicks to time spent on each page. This allows for a higher resolution analysis on student learning than using only results from the assessment questions or survey. In particular, this allows for the identification of significant moderating effects between the learning interaction data collected and the feedback. Finally, we use statistical methods to analyze the collected data, guided by rigorously validated instructional theories and pedagogies. In particular, we use hierarchical linear modeling to measure variability in LO performance simultaneously at both the student and LO levels.

These three aspects of our approach help us in answering four fundamental questions on using LOs for computer science education: (1) Do students learn from learning objects? (2) What attributes of the students and LOs are predictive of learning? (3) What is the effect of active learning, when a student can interact and determine the flow of the content? (4) What is the effect of feedback?

These questions are important to understanding how LOs impact student learning. The need for the first question is obvious. Identifying attributes that are predictive of learning allows revision of LOs and informs instructors of student behaviors linked to learning or lack of learning. In active learning, students actively interact with the LOs rather than passively reading the content. Previous work has shown that active learning generally enhances the learning process [1][9]. However, this has not been previously verified specifically for LOs on computer science. In feedback, students are given explanations immediately after they complete exercises in the LO. Varying the amount of feedback may have an impact on the learning process [13][21].

The remainder of this paper is organized as follows: Section 2 provides background on the instructional strategies and achievement variables used in the research. Section 3 discusses the LOs and our research project in more detail and discusses how the student interaction data is collected. Section 4 discusses the results and lessons learned from our research. Finally, we provide high level conclusions and future work for this project.

2. RELATED WORK
This section discusses previous work related to the questions addressed by our analysis.

2.1 Learning from LOs

Research has shown that students' use of learning objects increases achievement and promotes success [8][14][26]. While establishing the learning value of LOs is critical, it is important to go beyond this broad generalization and examine what specific characteristics of the students and LOs are predictive of learning, as well as to determine the effectiveness of LO-based instructional strategies such as active learning and feedback.

2.2 PREDICTORS OF ACHIEVEMENT

Here we review studies examining factors that are believed to function as indicators of achievement in an introductory computer science course. Comfort level, math background, and previous programming experience appear to be the primary factors under investigation across several studies [5][10][30][31]. Other factors include age, gender, intended major, self-efficacy, and motivation.

Some researchers have found comfort level to be the best predictor of success as measured by midterm course grade in a computer science class [31]. Comfort level refers to the set of physical or psychological circumstances in which students feel most at ease while asking questions in class, low level of anxiety while working on computer assignments, students' perceived difficulty of the course, completing assignments and understanding of the course concepts [31]. Students who perceived the course material as not difficult tend to perform better than their peers who consider the course difficult [24].

Math background has been shown to be important in predicting success [24][31]. Research findings suggest a statistically significant correlation between previous academic experience in mathematics and science to achievement as measured by continuous assessment and final grade [5]. However, other studies have found that the number of math courses a student took in high school did not affect achievement, and effort (as measured by percent of lab usage) and comfort level measures were more important predictors than SAT math scores [28].

Examination of previous programming experience has revealed mixed results. While some of the studies suggest that previous programming experience significantly influences achievement [31], others found no effect of previous programming experience on achievement [5][10][24]. Previous programming experience was found, however, to influence students' pre-self-efficacy [30].

Self-efficacy, or students' perceived judgment of their capabilities, appears to become more accurate over the course of a semester, and therefore, the timing of the measurement is important. Some students underestimate or overestimate their ability to perform. Their perception becomes more accurate as students learn to evaluate their abilities based on the direct interaction with the task [30].

Motivation is found to impact student achievement as well. Students with high intrinsic motivation perform significantly better in a computer science course than students with low intrinsic motivation [4][6].

Gender, age, and intended major did not have significant impact on achievement in a computer science class [3][5][28].

While it is important to keep in mind that the results are dependent on the settings, instructional materials used and the subjects, there are certain factors that seem to be more accurate in predicting successful performance in a computer science class and require further investigation. The results of our study confirm this. Student achievement was significantly correlated with average LO completion percentage (r=.56). Baseline self-efficacy was also significantly correlated with achievement (r=.30).

2.3 EFFECT OF ACTIVE LEARNING

Here we review learning theory and research emphasize that learning is enhanced by actively engaging students in the learning process [1][8]. Research has confirmed this principle with computer science undergraduate classes, showing that students gain greater mastery of the material and retain more information when active learning strategies are employed [1]. Active learning environments in computer science also impact student note taking, studying, and reading the textbook [12]. Other research has shown that the use of active learning techniques leads to better student attitudes and improvements in students' thinking [22].

Technology is a tool that can encourage active learning; computer-based multimedia instruction prompts students to become actively engaged with the material, making choices and actively manipulating objects and models on the computer screen. The term *interactive* is often used to reflect such computer-based active learning strategies. Our use of interactive learning techniques within our LOs was a critical instructional design consideration, and one aspect of our research focused on examining the effectiveness of this instructional strategy.

2.4 EFFECT OF FEEDBACK

Here we also review studies on the role of feedback in interactive exercises. Studies that report positive effect of feedback on achievement differentiate between various levels of feedback, such as verification (correct/incorrect), knowledge of correct response (correct answer) and elaborative feedback (addresses the answer and errors, provides explanation, example or/and guidance) [11][12][20][21]. Research on the effect of feedback on learning has generated inconclusive results. A meta-analysis of 250 studies conducted by Bangert-Drowns et al. [2] found weak effects of feedback on achievement.

Overall, researchers appear to be in agreement that elaborative feedback yields the highest scores [2][16][21]. More specifically, it was found that elaborative feedback produced the highest scores for low ability students, while verification feedback (correct/incorrect) produced the highest scores for high ability students [13].

Some studies indicate that elaborated feedback is significantly more effective on retention task than corrective feedback or general advice [16][20][29]. There is also evidence that elaborative feedback is more effective on a transfer task, since students that received elaborative feedback did better on post-test as compared to the pre-test [16][21].

3. DEPLOYMENT OF LOs

Our LOs follow the SCORM standard so they are usable on any SCORM-compliant learning management system (e.g., Blackboard, Moodle, etc.). Each of our LOs contains (1) a tutorial, (2) a set of flash/applet interactive exercises, and (3) an assessment. The tutorial starts with a page that lists the objective of the LO, followed by a set of pages that explains the content using text and graphics. The amount of information in each tutorial component is a succinct section on a particular topic—about several pages of a traditional textbook. The tutorial concludes with a summary and hints for reflections. Each LO also has a set of 1-4 exercises on the content covered in the tutorial.

Exercises generally require several steps to arrive at the correct answer. Students can repeat exercises as many times as desired. The assessment consists of a set 7-15 questions depending on the length of the tutorial. All the questions are either multiple choice or true/false. These questions are used to measure whether students understand the content presented in the tutorial and exercises.

We have developed LOs for all the introductory computer science concepts in the ABET approved syllabus for CS1 [25]. The LOs cover a comprehensive range of content ranging from basic concepts as arrays, numeric data, and logic to advanced concepts such as searching, sorting, and recursion. The LOs also cover a range of difficulty measured using both subjective and objective means. The subjective difficulty was determined using the vote of five content experts on a scale from 1-7 with 7 being the most difficult. The objective difficulty was computed using average student assessment scores. Table 1 shows all the LOs provided along with the subjective/objective difficulty.

Table 1. LO Content and Difficulty Rating.

LO Content	Subj. Difficulty	Obj. Difficulty
Advanced Logic	4.33	77.25
Advanced Recursion	5.33	71.14
Algorithms	3.66	77.74
Arrays	5.00	65.93
Conditionals	3.66	54.74
Debugging	4.33	75.27
Functions	3.66	80.78
Logic	2.33	85.59
Looping	4.33	63.06
Non OO Problem Analysis	5.33	74.73
Numeric Data	2.33	73.94
OO Problem Analysis	4.00	83.29
Recursion	4.66	68.01
Searching	4.00	85.74
Sorting	4.33	75.39
Variables & Constants	3.00	58.45

Active learning in our LOs is provided in the exercises. Exercises use a variety of response methods including drag-and-drop, fill-in-the-blank, etc. Students use multiple response methods to complete each exercise. We also provide a version of the exercises with no active learning. This “passive” version contains a video of the exercise being completed successfully.

Feedback in our LOs is also implemented in the exercises. Students receive feedback at each step during the exercise not just at the end. We provide two separate versions for the exercises with either verification or elaborative feedback. The verification feedback version informs the student when the step taken is incorrect and when the problem has been completed. The elaborative feedback version provides detailed explanations on why the student response was incorrect and also clarifies when the response was correct. For example, a verification feedback exercise on the Functions LO tells students that the parameters entered are incorrect (e.g., For the Arg2 box your entry of MakeLemonade is incorrect) whereas the elaborative version explains why the parameters are incorrect and provides hints on choosing the correct parameters (e.g., Arguments to functions are normally things, not actions. MakeLemonade is more likely to be the name of a function, etc.).

To collect the student interaction data, we make use of a software tool called Intelligent Learning Object Guide (iLOG) [18][23]. First, each LO contains a ***Wrapper*** that tracks all student interactions with the LO and uploads them in-real-time to an external database. The wrapper tracks not only scores on assessment questions, but also the time students spend on specific content, the steps taken on the practice exercises, etc. This provides data on whether students are struggling with the content that shows up later in the lecture. Second, the iLOG automatically administers surveys (as part of the LOs) to measure student demographic, motivation, and self-efficacy information. The survey results are used to generate individual student models used to improve analysis of the student interactions. The student interaction data collected using iLOG is summarized in Table 2. This includes static data collected from the surveys and interaction data collected from the wrapper. Finally, iLOG uses a ***MetaGen*** system to automatically “crunch” the data collected from the LO wrapper and student models. MetaGen provides details to the instructor on exactly what is causing students to struggle with the content. Such details are given as empirical usage metadata on what student interactions are linked with success or failure to understand the content.

The iLOG has been deployed at the University of Nebraska, Lincoln from 2007-2010. By the end of 2010, our LOs will have been used in 16 offerings of introductory CS courses involving over 1100 students. Table 3 shows the deployment details for our study. In all courses, the LOs were part of a student’s course grade. All the LOs together counted for between 3-5% of the total course grade based on instructor preference. This was done to ensure students had some motivation to take the LOs. The deployment schedule of the LOs varied between the courses to make sure students had the opportunity to take the LOs before the lecture/labs on the same topic. This was done to ensure student assessment scores reflected understanding of the LO content and not other sources (e.g., lecture on same content). As mentioned previously, all courses cover the same introductory computer science concepts. However, because of the different programming languages used (e.g., MATLAB, C, and Java) some revision of the LO content was required to accommodate underlying differences in the languages (e.g., Arrays in MATLAB are 1-indexed instead of 0-indexed). Also, the students in these courses included non-majors, CS majors, CS honors students as well as honors students in a special CS-business program. This required carefully balancing the difficulty of the assessment questions to accommodate students with varying aptitude.

Table 2. Student Interaction Data Collected using iLOG.

Static Student Data	Static LO Data	Interaction Data
Baseline motivation	Topic	Assessment scores
Baseline self-efficacy	Length	Avg. time per assessment question
Gender	Difficulty	Total time on exercises
Major	Feedback type	Min time spent on a tutorial page
GPA	Bloom's taxonomy for assessment	Avg. clicks on exercises
SAT/ACT score	...	Feedback received on exercises
...		...

Table 3. iLOG Deployment Details.

Deployment	# of LOs	# of Courses	# of Students
Fall 2007	3	1	~30
Fall 2008	8	4	271
Fall 2009	16	5	360
Spring 2010	5	1	48
Fall 2010	16	5	403
Totals	48	16	1112

4. RESULTS & LESSONS LEARNED

This section uses the empirical analysis on student interaction data to address four questions on LOs.

4.1 Question 1: Do LOs impact learning?

Our previous research has confirmed the learning value of LOs in computer science. First, Nugent et al. [19] compared achievement results for students using LOs to laboratory activities. We found that the LOs were an effective substitute for face-to-face laboratory activities and students rated them highly in terms of design, usefulness and appropriateness. Second, In a randomized experiment Nugent et al. [18] compared learning of students in introductory computer science courses who viewed LOs versus those that did not have access to these online resources (control condition). Results showed the LO condition resulted in significantly higher assessment scores than did the control condition. This previous work begins to show the efficacy for using LOs in CS education. However, we still need to determine (1) the specific student and learning object attributes that are predictive of learning (2) whether active learning and feedback has a significant impact on student performance. Both are addressed later—Sections 4.3 and 4.4—in this paper.

4.2 Question 2: What Attributes Are Predictive of Learning?

Our early projects showed that certain attributes of the student ―collected from the student interaction data―were more useful in understanding and diagnosing student success or failure. A major research component of our project has been identifying which of the many possible parameters are most useful to predict learning and learning progress. Regression analysis provides a way to evaluate the attributes of students and LOs that are significant predictors of individual performance in the assessment component of the LOs. Ordinary least squares regression assumes outcome scores are independent cases. However, students could have provided data for up to 16 LOs. Outcome scores that are from different LOs (but the same person) may be correlated due to clustering within students. Therefore, hierarchical linear modeling (HLM) was used to account for the clustered nature of the data (i.e. LOs within students). HLM accounts for this by estimating a variance component, in addition to the normal residual variance, that captures the variability in average outcome scores across students. In fact, a significant amount of the variability in LO performance was found at the student level. The HLM framework allows for predictors of LO performance to be assessed at the LO level (level 1; e.g. LO difficulty) as well as the student level (level 2; e.g. student gender). Table 4 shows several of the variables that have been identified as significant (p<.05) or marginal (p<.10) predictors of learning.

Table 4. Significant Predictors of Learning.

Student Attributes/Baseline Questions	B (p-value)
GPA	8.25 (<.001)
ACT Score	2.19 (<.001)
Number of Programming Courses Taken	4.42 (<.001)
Computer Science Placement Exam	2.02 (<.001)
Baseline Student Motivation	1.01 (.003)
Baseline Student Self Efficacy	1.52 (<.001)
LO Attributes/Evaluation Questions	
LO Difficulty	-1.04 (.083)
Assessment Total Seconds	-.01 (.001)
Assessment Total Clicks	.10 (.013)
"The learning object was easy to use."	5.93 (<.001)
"Was any part of this learning object confusing?"	-9.20 (<.001)
"Overall how would you rate this learning object?"	3.42 (<.001)

Entries in the second column of Table 4 are regression coefficients (**B**) and their associated p-values. The regression coefficients can be interpreted as the expected increase in percent correct for a one unit increase in the predictor variable. For example, a one unit increase in GPA (e.g. 2.0 to 3.0) leads to an expected increase of 8.25 percentage points on the LO assessment. Several LO- and student-level variables were significant predictors of student learning. The LO evaluation question: "the learning object was easy to use" is one of many LO evaluation questions that predicted student performance. Increased LO difficulty, increased time spent on the assessment, and reports of confusing LOs are also negatively associated with student learning and potentially negatively impact student learning. Student gender was not found to be a significant predictor.

4.3 Question 3: What Is the Impact of Active Learning?

Our research [18] has explored the impact of active versus passive engagement with the LO. In the active condition students manipulated graphical objects on the screen; in the passive condition a predetermined sequence of responses was demonstrated with no opportunity for students to interact with the material. Scores on the LO assessments were significantly higher for students in the active versus passive learning condition $F(1,390 = 4.62, p = .032)$. Supporting this result is our current finding that when students interact more with the LOs (measured by the amount of mouse clicks within an LO), their scores on the assessment are significantly higher ($B=.093$, $p=.019$).

4.4 Question 4: What is the Effect of Feedback on Student Learning?

Our research on feedback focused on two levels: (1) low level feedback as represented by simple knowledge of results, and (2) elaborative feedback, as represented by extensive explanations and models. Although the main effects for feedback were non-significant, there were several significant moderating effects [15].

Several variables were found to significantly or marginally moderate the effect of elaborative feedback, as shown in Table 5.

Table 5. Significant Moderators of the Effect of Feedback.

Student Attributes/Baseline Questions	**B (*p*-value)**
"I am confident in my computer science knowledge and abilities."	2.171 (.051)
"Compared to other students in this class I expect to do well."	2.224 (.068)
"After completing this course I expect to take more computer science courses."	2.064 (.038)
Total Self Efficacy	.421 (.100)
"The learning object helped me understand more about this topic."	2.266 (.050)
"I will use the same learning object again in the future if I have questions about this topic."	2.205 (.048)
"Overall how would you rate this learning object?"	1.870 (.060)
Number of Programming Courses Taken	1.929 (.089)

Elaborative feedback had a positive effect on scores among students with high ratings on a few of the motivation and self-efficacy questions. This is likely due to the fact that students were not forced to attend to the feedback. Those that were more confident and motivated may have been more likely to seek out feedback. Several LO evaluation questions also moderated the effect of elaborative feedback. Positive effects of elaborative feedback were found when students gave positive ratings to the LOs (e.g. it helped them understand the topic).

Instructional supports will not promote learning if students do not attend to them. Students spent an average of just under three minutes on the LO exercise sessions, which includes time spent reading feedback. The amount of time spent looking at feedback should be an important predictor of the effectiveness of elaborative feedback, but this was not tracked explicitly. The feedback was provided in a small window embedded low in the screen. In fact, students had the ability to skip the feedback completely, making it difficult to establish causal validity in the effect of elaborative feedback and is likely the reason that the main effect of elaborative feedback was not significant.

5. CONCLUSIONS & FUTURE WORK

Learning objects have been previously used in CS education either to support e-learning or supplement traditional classroom lectures. Previous studies focused only on assessment/survey results from a limited number of LOs. This made it difficult to understand how LOs impact student learning. We provide an empirical approach that addresses these issues by (1) deploying a comprehensive suite of LOs, (2) using a software tool to collect and store student interaction data, and (3) using statistical methods and instructional theories to analyze the collected data. This analysis provides the main contribution of this paper: key steps in answering four fundamental questions on using LOs for computer science education.

Do students learn from learning objects? We found that students using LOs had significantly higher assessment scores than the control group and that students rated the LOs highly in terms of usefulness and appropriateness. This supports our claim that LOs facilitate e-learning in CS education.

What attributes of the students and LOs are predictive of learning? We found that attributes significantly related to the assessment scores varied between the LOs. However, we found that several LO and student attributes were significant predictors of learning. Positive predictors included GPA, motivation, and self-efficacy. Negative predictors included LO difficulty and time spent on the assessment. Overall, such results indicate that struggling students learned less from LOs than high aptitude students. For our next study, we have revised our LOs to be more tightly coupled between each pair of components (tutorial, exercises, and assessment) to better link the content to improve student learning [7], with the hope that poor-performing students will be motivated to view the materials. We will also add a qualitative component to our study—an educational psychologist observing a sample set of students one-on-one on how he or she completes their LO assignments and interviewing them on their cognitive process—to identify why struggling students are benefiting less from the LOs.

What is the effect of active learning? We found that scores on LO assessments were significantly higher for students who had active participation with exercises in the LO compared to students who simply watched the exercises. Our results support the use of exercises with active learning in LOs on CS education. We now exclusively use exercises with active learning for our LOs.

What is the effect of feedback? We found in our initial study that the main effects of feedback were not significant. However, there were several variables with significant moderating effects. In particular, elaborative feedback had a positive effect on students with high motivation and self-efficacy. After our study, we discovered that students were not attending to the feedback on many exercises. In our next study we have made feedback more salient by requiring the student to at least scroll through the feedback before moving on to the assessment. The feedback will be presented in a new active window, allowing us to more accurately assess the amount of time spent reading the elaborative feedback as well.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 0632642 and an NSF GAANN fellowship.

REFERENCES

[1] Astrachan, O., Duvall, R. et al. 2002. Active Learning in Small to Large Courses. *Frontiers in Education, Annual* (Los Alamitos, CA, USA, 2002), T2A16-20.

[2] Bangert-Drowns, R., Kulik, C. et al. 1991. The Instructional Effect of Feedback in Test-Like Events. *Review of Educational Research*. 61, 2 (1991), 213-238.

[3] Bennedsen, J. and Caspersen, M.E. 2005. An Investigation of Potential Success Factors for an Introductory Model-driven Programming Course. *ICER* (2005), 155-163.

[4] Bergin, S. and Riley, R. 2005. The Influence of Motivation and Comfort-level on Learning to Program. (2005), 293-304.

[5] Bergin, S. and Reilly, R. 2005. Programming: Factors that Influence Success. *SIGCSE Bull.* 37, 1 (2005), 411-415.

[6] Bergin, S., Reilly, R. et al. 2005. Examining the Role of Self-Regulated Learning on Introductory Programming Performance. *ICER* (2005), 81-86.

[7] Boettcher, J.V. 2003. Designing for Learning: The Pursuit of Well-Structured Content. *Syllabus*. 16, 6 (2003).

[8] Bradley, C. and Boyle, T. The Design, Development, and Use of Multimedia Learning Objects. *Journal of Educational Multimedia and Hypermedia*. 13, 4, 371-389.

[9] Bransford, J.D. and Schwartz, D.L. 1999. Rethinking Transfer: A Simple Proposal with Multiple Implications. *Review of Research in Education*. 24, (1999), 61-100.

[10] Chen, C.S. 2002. Self-regulated Learning Strategies and Achievement in an Introduction to Information Systems Course. *Information Technology, Learning and Performance Journal*. 20, (2002), 11-23.

[11] Clark, K.A. 1993. The Effects of Feedback on the Problem Solving Ability of Academically At-Risk Students.

[12] Cordes, D. and Parrish, A. 2002. Active Learning in Computer Science: Impacting Student Behavior. *Frontiers in Education, Annual* (2002), T2A1-5.

[13] Hanna, G.S. 1976. Effects of Total and Partial Feedback in Multiple-Choice Testing upon Learning. *The Journal of Educational Research*. 69, 5 (1976), 202-205.

[14] Herrmann, N., Popyack, J.L. et al. 2004. Assessment of a Course Redesign: Introductory Computer Programming using Online Modules. *SIGCSE Bull.* (2004), 66-70.

[15] Kluger, A.N. and Denisi, A. 1996. The effects of feedback interventions on performance: A historical review, a meta-analysis, and a preliminary feedback intervention theory. *Psychological Bulletin*. 119, 2 (1996), 254-284.

[16] Moreno, R. 2004. Decreasing Cognitive Load for Novice Students: Effects of Explanatory versus Corrective Feedback in Discovery-Based Multimedia. *Instructional Science*. 32, 1/2 (2004), 99-113.

[17] Mundy, D.P. 2006. Using Learning Objects to Support Introductory Computer Architecture Education. *IASTED* (Puerto Vallarta, Mexico, 2006), 305-310.

[18] Nugent, G., Kupzyk, K. et al. 2009. Empirical Usage Metadata in Learning Objects. *FIE* (2009), 1285-1290.

[19] Nugent, G., Soh, L. et al. 2005. Design, development, and validation of a learning object for CS1. *SIGCSE Bull.* 37, 3 (2005), 370-370.

[20] Phye, G.D. and Sanders, C.E. 1994. Advice and Feedback: Elements of Practice for Problem Solving. *Contemporary Educational Psychology*. 19, 3 (Jul. 1994), 286-301.

[21] Pridemore, D.R. and Klein, J.D. 1991. Control of Feedback in Computer-Assisted Instruction. *Educational Technology Research and Development*. 39, 4 (1991), 27-32.

[22] Prince, M. Does Active Learning Work? A Review of the Research. *Journal of Engineering Education*. 223-231.

[23] Riley, S., Miller, L.D. et al. 2009. Intelligent Learning Object Guide (iLOG): A Framework for Automatic Empirically-Based Metadata Generation. *Artificial Intelligence in Education* (2009), 515-522.

[24] Rountree, N., Rountree, J. et al. 2002. Predictors of success and failure in a CS1 course. *SIGCSE Bull.* 34, 4 (2002), 121-124.

[25] Soh, L., Samal, A. et al. 2009. Renaissance Computing: an Initiative for Promoting Student Participation in Computing. *SIGCSE* (2009), 59-63.

[26] Van Zele, E., Vandaele, P. et al. 2003. Implementation and Evaluation of a Course Concept Based on Reusable Learning Objects. *Journal of Educational Computing Research*. 28, 4 (2003), 355-372.

[27] Ventura, P.R. 2005. Identifying Predictors of Success for an Objects-First CS1. *Computer Science Education*. 15, 3 (2005), 223-243.

[28] Whyte, M., Karolick, M. et al. 1995. Cognitive Styles and Feedback in Computer-Assisted Instruction. *Journal of Educational Computing Research*. 12, 2 (1995), 195-203.

[29] Wiedenbeck, S., Labelle, D. et al. 2004. Factors Affecting Course Outcomes in Introductory Programming. *16th Workshop of the Psychology of Programming Interest Group*. (2004), 97-109.

[30] Wilson, B.C. and Shrock, S. 2001. Contributing to Success in an Introductory Computer Science Course: a Study of Twelve Factors. *SIGCSE Bull.* 33, 1 (2001), 184-188.