

# A Learning Objects Approach to Teaching Programming

Victor Adamchik  
School of Computer Science  
Carnegie Mellon University  
Pittsburgh, PA 15213  
[adamchik@cs.cmu.edu](mailto:adamchik@cs.cmu.edu)

Ananda Gunawardena  
School of Computer Science  
Carnegie Mellon University  
Pittsburgh, PA 15213  
[guna@cs.cmu.edu](mailto:guna@cs.cmu.edu)

## Abstract

*The goal of this paper is to describe a new approach to a content creation and delivery mechanism for a programming course. This approach is based on the concept of creating a large repository of learning objects, each of which consists of the core material, code examples, supplementary notes, and review questions. A learning object will be uniquely described by a XML document and presents an interface for future search, retrieval and updating, as well as for potential connection to external assessment tools. Furthermore, we describe a new teaching, learning and authoring tool (called Adaptive Book) that will allow users to add new learning objects, modify current ones, and discuss concepts using a variety of representation models. With the adaptive book, an instructor will be able to design his or her very own course using a large repository of material, which will target a particular audience or a customized syllabus. In addition, the electronic adaptive book will serve as an interactive, continuously up-to-date learning environment for students. It will allow students to create personal learning profiles that are embedded into the core content of the course. We believe that in the near future, this electronic adaptive book will have the potential to replace or enhance traditional paper textbooks. In all physicality, it is impossible to have a personal instructor for each student, but with the adaptive book, we believe we are one step closer to realizing the dream of individualized instructions for each and every student.*

## 1. Introduction

One of the challenging aspects of teaching a programming course is how to provide the right information in the right context at the right time to

the right person. The introductory programming course (taught in Java) at our institution is made up of students who come from a variety of disciplines (information systems, business, biology, mathematics and engineering) and have different levels of programming experience. For example, some have taken C++ with object first approach and some have extensive programming experience in C or another procedural language. This audience has little conceptual understanding of multi-class programs, object oriented design, class methods, parameter passing, and inheritance. The first challenge of this semester long introductory programming course is to bring all students to a common learning environment within 3-4 weeks.

One of the major problems of bringing students to the same level of understanding in a short period of time is the lack of an effective communication mechanism between instructor-student and student-student to share crucial knowledge at the right time. Students often misunderstand concepts and thus apply them incorrectly; which leads to hours of wasted time spent on debugging logically incorrect code. By the time students get help it is either too late or there is little opportunity for the instructor or TA to intervene. Our online surveys have indicated that students are reluctant to seek clarification about a concept instantly because most students do not have an adequate background to accurately compose a programming question through email or using a discussion board. Although course management systems (CMS) such as Blackboard, WebCT and eCollege include communication tools, their discussion forums are not effective since the discussion itself is not imbedded into the learning material. The communication tools of the aforementioned systems do not allow students to discuss a programming question in its own context. For example, a student cannot highlight or annotate specific passages in the textbook (or core content) or supplementary materials and initiate a discussion

thread around it. Current course management systems are excellent tools for transferring information but not knowledge.

Another problem which we identified is that textbooks are severely under utilized, since they do not provide easy and effective ways to interconnect major components of the course such as code examples, and review questions to the core content. Students in a course are generally not capable of sorting out all information and figuring out what makes sense to them. Clearly the current use of textbooks in programming courses is not effective because static books do not allow any kind of content personalization, enhancement or interaction.

We believe that we can address some of these problems by creating a better knowledge transfer mechanism between the teacher and the student. We will focus on two aspects:

- presenting key programming concepts by utilizing learning objects;
- establishing an interactive communication platform for discussions and exchange of learning material.

Addressing the first aspect, we need to make sure that all course related material is organized into a repository of information objects. The key components of the repository are textbook content, code examples and review questions. Then instructors can create learning objects using the repository of information objects; and share them with colleagues and students. In the long run, this will lead to a library of learning objects to which instructors can make contributions, as well as use publicly available learning objects in their own courses.

Addressing the second aspect, we need to make certain that all knowledge components are assembled under one unified and interactive environment. We will achieve this by creating an adaptive book toolkit. This tool will serve in various ways, including a) it will allow instructors to create and share customized content to meet the learning objectives of individuals or groups; b) it will allow students to create personalized learning profiles and share them with others; c) it will create an environment where students can discuss course-related material in its own context.

## 2. Learning Objects

We define a learning object (LO) as an integrated module containing the core text, code examples, review questions, supplementary material, and programming labs. A learning object is uniquely

described by an XML document that includes metadata and semantic relationships between LO components. The XML document also serves as an interface for future search and retrieval of LO's. Moreover, the LO will provide an open interface for a connection to other components, such as external assessment engines. Each LO will be developed according to IMS/SCORM [1] standard, which will assure the interoperability and reusability among all learning platforms. SCORM is an emerging standard for creating, sharing and extending learning objects independent of the platform or the audience.

Each LO is editable and can be tailored (by creating a learning profile) to meet the needs of an individual student or a group of students. Learning object profiles (LOP) then will be distributed to students as email attachments or can be placed in a course management system for easy downloads. Learning object profile is a relatively small XML document that describes the components within the learning object. This provides the benefit of low bandwidth downloads as well creating LOP's that describes a guided tour within the digital content. We believe that the LO approach is the first step toward achieving personalized content which addresses the learning needs of each and every student.

Developing a comprehensive library of LOs to support individualized learning needs is one of the long-term goals of this project. We are currently studying the LO tagging mechanism to find an easy way to search and retrieve the LO to fit a particular learning need. We also expect to create a mechanism for users and developers to evaluate LOs and to classify them according to specific criteria. There are many open questions in developing learning objects. What is the correct size of a LO? A too big or a too little LO may not well serve the learning objectives of the module. How do we address issues such as reusability, platform independence, and extensibility of a LO? These questions depend not only on standards but also on the discipline. We will address these questions as we continue to research and gain more insight into the subject matter.

In the long run, we plan to create a library of LO's that will be able to fit the needs of an entire class or an individual student. Although it seems difficult to meet the individual learning needs of every student, we will be able to use the interactive tools to create and maintain a library of individualized LO's. In the future, we plan to use classification engines such as Vivisimo developed at Carnegie Mellon, to automatically search and cluster learning objects as a response to a specific question. This will allow students to automatically search for

LO's to meet their individual needs. We strongly believe that over time we will be able to achieve the goal of creating an individualized library of LO's for teaching and learning computer science.

### **3. Adaptive Book**

In this section we will discuss the notion of the Adaptive Book (AB), a platform under which LOs are developed, distributed, and discussed. The Adaptive Book serves as a browser for interactive content as well as an authoring environment for creating and managing LO's. Unlike typical course management systems, AB allows creation and transformation of knowledge (not just information) under one unified and interactive platform.

The AB has an interactive toolkit that allows instructors to customize and thus create their own digital course content targeted towards a particular audience or an individual. It also provides special tools for the instructors to author and link supplementary material. The AB browser allows users to add bookmarks, highlight and annotate text, add active hyperlinks, attach text and binary files, and incorporate questions and answers; ultimately, it will have a copious amount of interactive features built around the core content of the course. The AB is the first software application that merges the process of teaching and learning into one system. For example, an instructor can select a specific chapter or section in the core content and create a learning profile to fit the needs of a particular student or the whole course. The students will then be able to download this profile and integrate its content into the digital book. These profiles could be as specific as a guided tour through the content, or as small as a highlighted section or an annotation. We believe this level of specificity allows students to master programming concepts quickly and increase the efficiency of communication between instructors and students. The present state of AB technology allows creation, distribution and enhancement of LO's. AB is under further development to better facilitate authoring and distribution of learning objects created according to the IMS/SCORM standards.

### **4. Pilot Project**

During the summer of 2002, we taught a second semester programming course with an introduction to data structures such as arrays, stacks, queues, graphs and trees. Java was the primary language used in this course. Students taking this course had basic programming skills in procedural languages, mostly

C and Basic. However, the adjustment to the rigor and difficulty level of programming courses taught at Carnegie Mellon is a great challenge for majority of the students. In a short period of time (say 1/3rd of the semester), students have to learn a new programming language as well as learn how to use the language to understand new object-oriented programming concepts. This first third of the course is a fundamental part of the course that influences the transition to data structures. As we saw during the summer pilot program, students were overwhelmed by the language and style differences of the course. Providing lecture notes in advance was not very effective, and extending office hours was not proven helpful. Clearly, the audience of this course required alternative teaching and learning methods. The main problem as we noticed was that students were not able to absorb all of the required material in a short period of time - language proficiency requires time. Preaching that an object can be created out of other objects either by composition or inheritance will leave only a slight trace in a student's mind. Such fundamental concepts can only be understood through repeated coding exercises performed by students themselves. Moreover, these code examples must be linked to a particular section or paragraph in the lecture notes, and must contain an abundance of optional material that will help them learn concepts faster. Another problem, according to the students surveyed, was that there were significant differences between reading the lecture notes and reading the textbook. Students liked to have a short, annotated, and highlighted version of the textbook that will allow them to quickly find the information they need. This is somewhat similar to notes students are taking in class – notes that make sense only for those who attended the class. This challenged us to rethink the way we deliver material to students. We focused on the fundamental aspects of a “lecture” and started thinking of ways to reinvent it. We redefine “lecture” as a composition of learning objects, where each object consists of specific highlighted and annotated content, code examples as well as quick pre-tests to make sure that students possess the skills necessary to understand the material. We believe that less note taking in class helped students to concentrate more on class discussion.

The next question we ask ourselves is what are the lecture notes? This definitely should be larger than just a collection of learning objects. Students need an environment where they can discuss course-related content in its own context. It is much more beneficial for students to annotate a specific paragraph or code example and ask questions within the content rather than sending a question or an entire

program to the instructor or TA via email. Our online surveys indicated that even if the student did not understand the discussion in the textbook, they were reluctant to seek clarification of a concept. This is because most students could not accurately articulate a programming question using an email message or using a discussion board. We found that less than 10% of the class actually participated in public online discussions using a CMS. We believed that we could reduce the communication overhead and allow students to conduct more effective online discussions on programming assignments if we create an environment where communication can happen in the context of the course material. Elaborating on this idea led us to the adaptive book - an interactive content display environment as well as an authoring tool for creating and managing learning objects. One important component of this model is that an instructor can create a customized learning object and share it with an individual student, a group of students, or the entire class. For example, using the authoring toolkit of the adaptive book, an instructor can create a guided tour within the book during the class lecture that captures the essence of the lecture. Students then will be able to download these tours from a repository (such as a CMS) and directly integrate them into the textbook or core text content and view or personalize content to fit their learning needs. Mayer and Gallini demonstrated the effectiveness of captioned illustrations (see [2, 3]). They concluded that students who read a text containing captioned illustrations placed near the corresponding words generated about 65% more useful solutions on a subsequent problem-solving transfer test than the students who simply read the text. Instructors can create these illustrations fairly quickly using the adaptive book's authoring toolkit. The second aspect of the communication model was to provide students with a flexible learning environment to quickly transfer their code into a communication platform and allow them to highlight and annotate and share that with instructor or fellow students. We achieved some of these goals in the second semester programming course we taught during the summer of 2002. Students generally liked the idea of receiving specific LOP's. The highly focused nature of a LOP allowed students to quickly understand the concept and practice programming skills. Students were also able to exchange LOP's among themselves and hence creating a collaborative learning environment. This was our first attempt to make the delivery and management of the course more efficient and effective and the post-course surveys indicated an increased level of enthusiasm for learning in the environment.

## 5. Conclusion

There are many areas within the AB environment that are worth exploring. For example, there is significant promise in further developing the guided tour concept and refining the online discussion thread in the context of the course content. We need to develop a better mechanism to create learning objects, share and evaluate them. We also need to further study the user behavior in the context of a digital textbook and supplemental content. In fall of 2002, a group of graduate students from Human Computer Interaction (HCI) Institute at our university have been studying this problem from a HCI point (see [4]). The study involves test subjects from introductory programming courses and building a better user communication model based on the multiple representation principle (see [2]). This interdisciplinary group of HCI students (consisting of computer science, design, engineering and psychology majors) will expand the results of the summer pilot. Another focus is to develop algorithms to automatically evaluate LOP's that are created as part of the learning process. This new integrated online learning environment generates massive data sets and using data mining to analyze the data is part of a long-term project. In summary, it is quite interesting to develop a learning model where all content used in a programming course is tightly integrated into a unified framework and all authoring, distribution and discussion occur in that context. We believe that the learning objects based teaching method will significantly increase the pedagogical effectiveness and the context-based communication effectiveness over traditional methods such as email and discussion forums.

## 6. References

- [1] Advanced Distributed Learning Network, <http://www.adlnet.org>
- [2] R.E.Mayer, "Systematic thinking fostered by illustrations in scientific text.", *Journal of Educational Psychology*, **81**(1989), pp.240-246.
- [3] R.E.Mayer, J.K Gallini, "When an Illustration worth thousand words" *Journal of Educational Psychology*, **82**(1990), pp.715-726.
- [4] N.Moraveji, A.Travis, M.Halpern, M.Bidinost, A.Gunawardena, "Designing a User Interface to facilitate communication in the context of a digital textbook", HCI Institute project report, Carnegie Mellon University (2002).