Designing SCORM Compliant Courses for Introductory Programming Students

Michael DiBlasio, Giovanni Vincenti, James Braman

Towson University
Dept. of Computer and Information Sciences
8000 York Rd. Towson, MD. USA
mdiblal@students.towson.edu
{gvincenti, jbraman} @towson.edu

Abstract
In this paper we present our preliminary work and outline towards the development of reusable SCORM (Sharable Content Object Reference Model) compliant courses to be deployed in the introductory programming curriculum. As many students often have difficulty understanding key programming concepts, various approaches of learning data structures and programming elements can be helpful. We describe the rationale of the project in terms of computer science education followed by a description of Learning Objects (LOs). Next we describe SCORM and the benefits of using such a model, our current work followed by an outline of our next steps and goals. Through this project we hope to begin to realize new ways of expressing material for students starting to learn how to write programs.

Introduction
The United States Bureau of Labor suggests that Software Engineering and Computer Programming occupations will rapidly grow from 2009 to 2018 (2010). Because of this fact, Computer Science as a major is becoming increasingly inviting to students that typically would not think about entering the programming world. Typically the first experience many Computer Science undergraduate students experience is an introductory programming class which often emphasizes key concepts and programming basics using a particular programming language. While some students perform exceptionally well in these introductory courses, there are many that quickly fall behind and often do not fare well. Bennedsen and Caspersen’s study reveals that failure rates in 54 surveyed US institutions averaged at 33% for some introductory programming courses (2007). One in every three students in their study had problems understanding fundamental elements of programming. In addition to this study, Georgia Tech reported that the combination of the number of students withdrawing and that of failing students for most introductory classes ranged between 30-50% (Guzdial and Ericson, 2010). At Towson University, students entering Computer Science I need to pass a small programming assessment to gauge if they have grasped the necessary foundations for this first main computer programming course required by the major (Dierbach et al, 2005). If a student receives a low score (or if a student based on a relatively low score does not feel they should proceed with the class) the faculty member recommends them to take more practice courses to grasp key concepts. Most often these are true introductory programming courses that precede major course work. Due to these typical findings and common pitfalls associated with entry level programming courses, scholars have examined alternate forms of teaching methods. Many difficulties plague introductory programming students from grasping key concepts. While many techniques have been experimentally explored by many instructors, the problem still remains and many student have difficulties. As programming is an essential skill for many computer related occupations, increasing student knowledge is very beneficial in the long term. In this project we explore introductory programming education through the use of Learning Objects within the context of the development of SCORM (Sharable Content Object Reference Model) compliant Learning Objects (LOs). The goal of SCORM compliancy is to maintain an aspect of cross-platform and cross-device independency so students can use these learning tools on many devices as well as having the capability to easily modify objects for new content.

Learning Objects
One way to enhance student’s understanding of basic concepts is through the use of Learning Objects. Learning objects are versatile digital learning artifacts that can be used independently or in addition with traditional teaching methods to help students understand certain concepts. Our focus with these learning objects is within the context of introductory programming basics. As an example of its use, the
London Metropolitan University Staff created and used learning objects in multiple introductory courses to increase student performance (2005). Their results showed that some classes saw as much as a 27% increase in pass rates with over 92% of students stating that the learning objects were useful or very useful (London Metropolitan University, 2005). The Learning Object Metadata (LOM) standard defines a learning object as “any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning” (LTSC, 2005). These objects can be based on various forms of digital technologies such as web based, standalone applications, or based on interactive 3D objects (Sullivan, 2009). Learning Objects can be used in conjunction with Personal Learning Environments to provide students with a data-rich personalized learning experience.

**SCORM**

Sharable Content Object Reference Model (SCORM) is a “Collection of specifications adapted from multiple sources by Advanced Distributed Learning (ADL) to provide a comprehensive suite of e-learning capabilities that enable accessibility, interoperability, durability, reusability, and cost effectiveness of Web-based learning content” (Fentress, 2004). Advanced Distributed Learning details that using SCORM is beneficial because it allows one to be able to locate and access instructional components from multiple locations and deliver them to other locations, and to take instructional components developed in one system and reuse them in another system (ADL, 2010). They also mention that the technology can withstand evolution and/or changes without costly redesign, reconfiguration, or recoding, and provides flexibility to incorporate instructional components in multiple applications and contexts. This system was developed by the Department of Defense to “Ensure that the learning experience and performance data tracking is consistent in the distribution of training courses via the Internet, and allows for online collaboration between users” (DOD, 2009). Because of its success in multiple disciplines, the U.S. government is increasingly making SCORM a requirement. In addition to this, as SCORM gains popularity, Fortune 500 companies are beginning to adopt SCORM and government grants are beginning to require it (Fentress, 2004).

**Overview**

As our model focuses on introductory programming concepts, we are designing SCORM-compliant LOs that focus on control basic programming concepts initially, such as IF-Statements, While loops, For loops and Arrays. Figure 1 highlights these four main modules and their relations. Once a student has successfully progressed through one module, then they can proceed to the next concept. Students are allowed to progress backwards to reinforce material if needed, but not allowed to move forward until they have successfully demonstrated their mastery of a particular skill through built-in assessments.

A similar approach is underway with the development of modules that can teach basic SQL (Structured Query Language). Figure 2 highlights the main components of common SQL statements: CREATE, INSERT, DELETE, UPDATE, SELECT and JOIN.

**Practical Applications**

We wish to include the learning tools described in this article as an active component of Computer Science courses at Towson University. Our department includes three programs: Computer Science, Information Systems, and Information Technology. Each program takes a different path, but all students are required to take programming courses. One course that is strongly encouraged as a first programming course acts as an introduction to computational thinking, focusing on pseudocode and scripting languages (Dierbach et al, 2005). The second course, which should be taken only after the first course is completed successfully, is based on the Java programming language and addresses material from compiling programs all the way to the introduction of object-oriented programming.

The program then offers a variety of courses in many other subjects within the domain of computing. A third course in which most students participate addresses databases. This course also acts as an
introduction to this topic, spanning from concepts of databases compared to file processing all the way to the implementation of programs that actively interact with databases.

In our opinion the two courses generated by the aggregation of SCOs described in Figures 1 and 2 would create a complete companion to the material discussed in each of the courses we briefly introduced. In the case of the programming courses it is clear that the SCOs follow quite closely the evolution of the concepts through the entire semester, allowing instructors to insert in each SCO a set of assets that can be utilized as lecture or laboratory material. The ability to stretch the online companion through almost the entire duration of the course allows for a return of investment in time and resources invested quite favorable. The aggregate that relates to databases instead offers a less complete coverage of the material discussed during the classroom and laboratory time. Typically, students spend one week working on the data definition language (DDL) and two weeks on the data definition language (DML). When working on DML we typically break the content in one week dedicated to statements on a single table and the second revolves around querying information from two or more tables. This breakdown allows us to discuss Join operations in more detail. At a first analysis the return of investment for this particular course is less conspicuous if compared to the extent of the programming part. For this reason it is crucial that the assets utilized for any database course applying the approach introduced in this article should be particularly effective in simulating database operations and assessing the proficiency of each student. Through this approach we believe that the students will be able to navigate through the content more easily, increasing the retention of the information introduced both in class and through the online component.

Future Work
We will incorporate the material introduced in this paper into our courses as early as next semester. It is our goal to increase retention of key materials in the context of introductory programming and database courses by using the solutions outlined in this paper.

Based on our initial assessment and preliminary feedback obtained by during the development period our next goal is to implement effective learning objects that can be incorporated into our courses as assets. Through more rigorous evaluation and targeted in-class assignments during the next implementation phase, we will collect more detailed data about the effectiveness of these tools. Through questionnaires and comparisons to other courses where we will not subject the students to the use of these learning objects we will then gauge the potential benefits.

As a last step to this pilot project we wish to evaluate the validity of our approach as it parallels traditional in-class instruction. The result of this analysis will let us spring into a deployment of full learning objects and technology-based instruction material such as video and audio presentations. As a logical consequence we will explore the applicability of this methodology to other lower level and upper level undergraduate courses in the disciplines of Computer Science, Information Systems and Information Technology.

Conclusion
With this project we have discussed our preliminary model of the development of reusable SCORM compliant learning objects for the use in introductory programming and their potential use in other related courses. We have also outlined the role of SCORM and the model of our learning objects. As we begin to refine and further develop this project into a more effective tool, it is our hope that compliant and reusable learning objects can be used for web-based learning among other things. Because of the difficulty often attributed to first-time students learning these concepts, our intention is to improve the retention of such materials. As with all learning tools, once enhanced and enriched with added material, and widely adopted, students ultimately will benefit.

References


