Proposer: Roxanne L. Canosa, Axel T. Schreiner, Dustin E. Mulcahey
Department of Computer Science, Rochester Institute of Technology
134 Lomb Memorial Drive, Rochester, New York 14613
phone (585) 475-5810, fax (585) 475 7100
mailto: {rlc, ats, dem5302}@cs.rit.edu

Topic: Web-based editable Java tutorials

Relevance: An important component of programming language learning is reading code to acquire idioms which facilitate expressing one's intentions. Understanding the effects of a program is helped if the output of the program can be studied together with the code.

The tutorials described here go a significant step further than printed pages: on a web page they provide Java programs (or fragments) and explanations, and they suggest modifications to the code. The student edits the program right in the web page, submits it to a web server for compilation, and receives it back as an applet for execution in the browser.

The approach allows for a very interactive mix of description, code, and modification tasks, and requires no installation or download on the client side. The programs have no access to the client machine and are not aware of their web-based compilation or execution.

Content: A highly interactive electronic "textbook" was created and used as a learning aid in an introductory computer science course. The textbook consisted of a set of web-based tutorials that were developed with the intent of helping students understand the process of programming, along with helping them to learn language syntax and program structure.

The tutorials were built around a web server that compiled Java programs and returned the result to the client as an applet. The tutorials included an editing environment that allowed students to test programs and experiment with code examples embedded in the text, without the need to install or run the Java SDK on the client machine. The service is also useful for enabling rapid prototyping with Java-related compiler tools without the need to install those tools first.

The poster describes the architecture of the web compilation service wcs [1], explains how it was used as a learning aid in an introductory programming course [2], and determines its effectiveness.

Wcs implements a Java-based compiler on a web server and executes the code in an applet. All of the software development tools reside on the server; the client only needs to execute unsigned Java applets. Wcs accepts source text for small Java programs and compiles the source. The resulting class files are merged with runtime support, archived, and sent to the client as an applet, where the program is executed.

The service is implemented as an HTTP servlet which processes GET and POST requests. The request parameters are designed to fit HTML form elements. The requests originate from the client side via a web page with one or more textarea
elements that serve as a trivial editing environment for programs. The tutorials consist of a set of web pages structured around initialized textareas that contain Java source code. The text on the web page is a detailed discussion of a selected topic, and suggests changes that the student can make to the source before submitting the program for compilation and execution.

The initial proof of concept for wcs was to implement a fairly extensive tutorial on language processing [3] which has been used successfully in courses on programming language concepts and compiler construction. More recently, wcs was used as a learning aid in an introductory computer science course via a set of interactive web-based tutorials [2]. The tutorials are intended to help students who are new to programming learn basic Java syntax, program structure, programming methodology, and problem-solving techniques.

An informal study was conducted to gauge the effectiveness of the tutorials as a learning aid: one version used wcs to supplement the text with an interactive demonstration of the topic, and another version only provided the same textual information about the topic along with static source code for examples. Students were randomly assigned to one of the versions at the start of the study and instructed to visit the website for their assigned version each week, when a new topic was posted.

The study provided evidence that students enjoy being able to experiment with code that is embedded in a tutorial explanation of the concepts and techniques, and perform better on quizzes testing programming skills. This suggests that this kind of learning aid is useful for imparting programming skills and engaging students in the programming process.

The development of wcs was supported by a Provost’s Learning Innovation Grant from the Rochester Institute of Technology.

References: