Teaching Students Programming:

A proposed web-based Java CASE tool with disciplined software development process

Dr Ashley M. Aitken

A.Aitken@Curtin.Edu.Au

School of Information Systems

Curtin University of Technology

Perth GPO Box U1987 Australia

Abstract

The focus on programming concepts and programming language details in introductory programming units is often at the expense of any sort of disciplined software development process. It is understandable then that students develop poor software development processes (when they are trained during this initial stage to focus primarily on coding). This paper describes a proposal for a Web-based and much simplified Java CASE tool that leads students through a disciplined software development process as they learn to program. The system will also enable the automatic collection of data about students' software development process, will assist in the on-line marking of student programming exercises, make it more difficult for students to plagiarise others' work, will remove the need for complex integrated development environments, and allow additional instruction on software development process to be provide directly to the students as they develop programs.

Introduction

Programming vs Hacking

Although we try desperately to encourage students to follow a disciplined software development process (including planning, requirements, analysis, design, specification, coding, and testing) in their programming the reality of the situation is that for most University programming exercises most students hack! In particular, this means they generally complete the exercises by using a brute force approach (such as repeatedly coding and compiling) focusing only on the code and without a disciplined engineering approach to software development. There are two factors which it would seem contribute the most to this situation. The first is the restrictions on time generally associated with University courses that result in the programs that students are asked to write being generally small and "hackable." The second is the use of commercial integrated development environments (IDEs) and compilers that generally encourage students to follow the code-compile-code-compile loop so indicative of hacking.

Traditionally, introductory programming units (in Computer Science, Information Systems and other departments) have focussed primarily on language and technical concepts and secondarily (often incidentally but sometimes more thoroughly) on the process of software development. At best these courses discuss the problem-solving nature of programming and at worst they focus purely on language specifics. This paper suggests that this is almost completely the opposite of what should be happening. The primary focus should be on the process of software development and the secondary focus on language and technical concepts (that, of course, are still essential to the task). In a similar way, we are seeing a redirection of emphasis today in some professional software engineering companies, where complex programming languages (such as C++ (Stroustrup 1986)) are being discarded for simpler languages (such as Java from JavaSoft at www.javasoft.com) to allow a more concerted focus on process instead of a continual fight with the technicalities of the programming language.

Overview of Paper

This paper will outline the design of a Web-based system for teaching students a programming language and programming language constructs within a discipline software development process (or as we suggest, one should consider it the other way around). The focus on a disciplined software development process is a result of the work on the Personal Software Engineering Process (PSP) by Watts Humphrey (Humphrey 1995; Humphrey 1997). In section two, a number of goals and constraints on teaching programming and software development to students (mostly within introductory level University units) are discussed.

In section three, a means to achieve these goals within the specified constraints is described in the form of a Web-based system for teaching software development and programming. In section four, the novelty of this proposed approach and some relevant educational issues are discussed.

The Challenges of Teaching Programming

Two of the primary goals of teaching programming to students are:

- To teach students the fundamentals of computer programming in an appropriate programming language, and
- To develop, within students, an understanding of a more complete software development process than that afforded by a simple coding focus.

The first goal is generally met admirably by introductory programming courses at academic institutions (and elsewhere). The important aspect in any programming course is to not over-emphasise the programming language aspects but, instead, to focus on a solid understanding of fundamentals of computer programming. The particular language is not of critical importance, it is the programming concepts that are important.

The second goal is, unfortunately, is not as well-addressed in most introductory programming courses at academic institutions (and elsewhere). Again, this is generally not through any lack of trying or aversion to software development process. As discussed at the beginning of this paper, usually the lack of time and appropriate tools acts to limit success with this aspect of education in software development.

The Constraints

Along with the goals mentioned above, there are a number of constraints imposed upon teaching programming and software developments to students in academic institutions (and elsewhere). These include:

- *Ability to only set short programming exercises.* Due to the fact that students are generally required to cover a large amount of new material in a short amount of time, it is usually not possible to make programming exercises large enough so that a non-disciplined approach to development would stand less chance of being working effectively.
- *Inability to use complex development tools.* Again, due to the lack of time (and also a desired not to overload the students), it is generally not possible to use complex Computer-Aided-Software-Engineering (CASE) tools, or even complex Integrated Development Environments (IDEs) (even if these were appropriate). Any tools need to be simple and functional without an excess of features that can confuse or distract students.
- *Lack of resources to supervise students' work.* Due to a reduction in funding it is generally becoming more difficult to provide a high level of supervision for students in the laboratory components of a programming course. Hence, students are required to work somewhat on their own with only the aid of practical worksheets or similar. It is difficult to ensure that students follow a disciplined approach to software development if they are not well supervised.

Much has been written about which is the best programming language and programming paradigm for teaching programming (Brilliant and Wiseman 1996). This, however, is not the focus of this paper. There are other approaches to teaching programming (eg see (Biddle and Ewan 1998) which focuses on teaching programming through reusability), even some that use multi-media (Wolz, Weisgarber et al. 1996).

However, there is nothing that the author is aware of that describes tools or systems to assist in emphasising good software development process in introductory programming courses.

An Environment for Teaching Programming with Process

This paper describes a practical way of achieving both these goals within these constraints. It proposes a Web-based (and much simplified) Java CASE tool that forces students to follow a disciplined software development process whilst learning programming concepts and a programming language. It replaces the need for a commercial IDE or limited functionality command-line tools and at the same time provides valuable information back to the students and the instructors on the student's performance (with regard to the software development process).

System Description

The proposed system (in its simplest form) consists of five main components:

- A sequence of Web pages which include directions and input fields for students to follow in the process of developing their program,
- A Web-based application that takes the inputs from these Web page forms, applies certain constraints to the inputs, constructs a Java program from these pieces and finally compiles the Java program,
- A database for storing the students' programs and process data (ie data describing what the students have done, eg time in each step)
- A Java-capable Web browser that can run the compiled (bytecode) Java application (applet) when it is downloaded to the student's web browser
- A Web-based application for reporting on the data collected about the students' software development process (eg how many iterations they performed within the development lifecycle)

As well, the system could include the following additional components:

- A Web-based application to allow staff to run and mark the student's programs along with their software development process data
- A Web-based application to assist students in estimating and planning the effort required to develop the required program (possibly based on the sum of their previously collected process data)

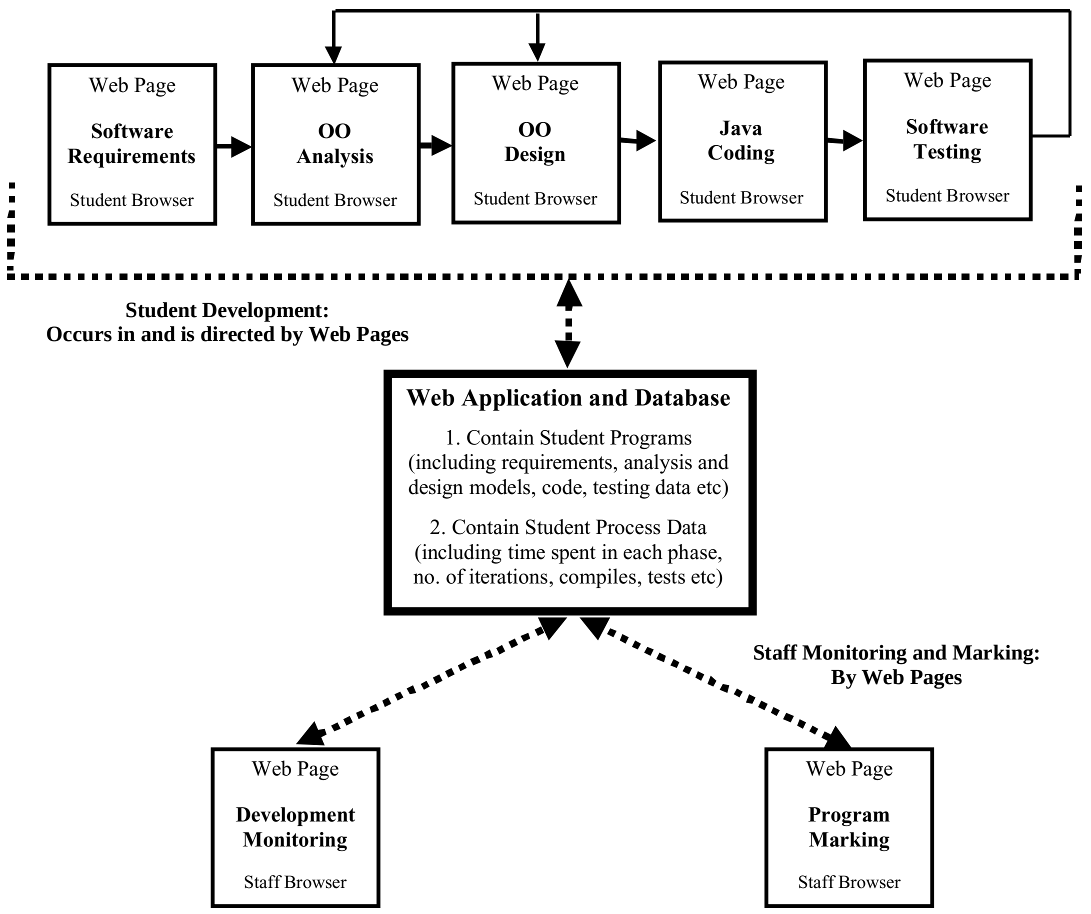
Web Page
Software Requirements
Student Browser
Web Page
OO Analysis
Student Browser
Web Page
OO Design
Student Browser
Web Page
Java Coding
Student Browser
Web Page
Software Testing
Student Browser
Student Development: Occurs in and is directed by Web Pages
Web Application and Database
1. Contain Student Programs (including requirements, analysis and design models, code, testing data etc)
2. Contain Student Process Data (including time spent in each phase, no. of iterations, compiles, tests etc)
Staff Monitoring and Marking: By Web Pages
Web Page
Development Monitoring
Staff Browser
Web Page
Program Marking
Staff Browser

Figure 1. Student Directed CASE tool (including Staff Monitoring and Marking)

Three-Tier System

The proposed system will be a three-tier web-based application (that can be more generally be seen as a web-based service). The first tier will be the Web browser that provides the user-interface or presentation layer and possibly includes some JavaScript or Java Applets to incorporate validation of inputs or other application logic on the client side. The second tier will be a Web application server running the logic of the disciplined development process on the Web server. The third tier will be a database server that will be used to store all the students' program and process data.

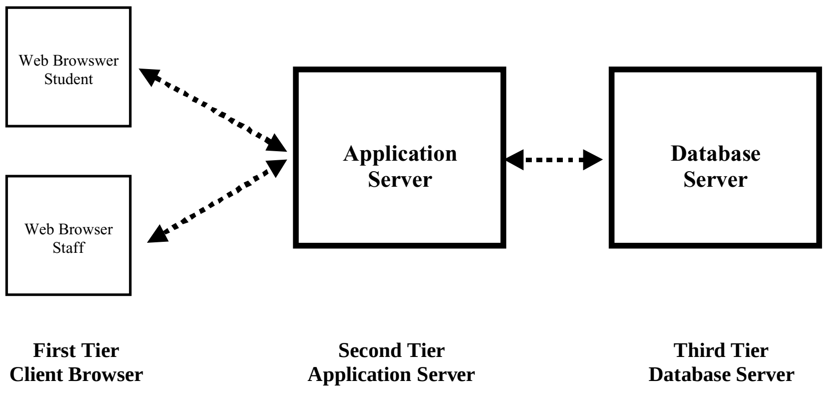
Web Browswer Student
Web Browser Staff
Application Server
Database Server
First Tier Client Browser
Second Tier Application Server
Third Tier Database Server

Figure 2. Proposed Three-Tier System

The benefits of a web-based three-tier system are significant. Firstly, being a web service the system can be accessed and used from any system with Internet access and a suitably capable web browser. Students can do their programming exercises at University, at home, and elsewhere, with little or no need for the installation of special software. Secondly, the system enables any suitable database server to be used (since it is somewhat separate from the application server). A University could use a large enterprise database management system whereas a personal installation could use a smaller personal database management system.

Process Steps

Central to design of the system proposed here is its ability to direct (or lead or force) students through a good software development process and provide educational instruction at appropriate stages through the process.

Students will develop their programs incrementally across a series of Web pages that lead them through a disciplined iterative and incremental software development process. Each step (from page to page) can have a number of checks and balances placed on the student's input and can produce output for the next step in the software development process. Instead of just being code-compile-code-compile, the Web application will direct (or lead or force) students back to earlier analysis and design pages for any significant changes.

Additional information can also be placed on the Web pages to assist students with the software development process. For example, on the page associated with design, additional information could be provided to assist students with the design process (eg principles of good design or tips and techniques for design). Such information can also be tied to the level of accomplishment of the students with their software development. New students could be given introductory advice and guidance, whereas advanced students could be given more advanced advice and guidance.

As an indication of how the system shall work, consider the following sketch of the Web pages used to develop a simple object-oriented program using a disciplined software development process:

- The first web page can contain a form for the student to enter a description of the problem (ie a problem statement). It may also allow for specification of functional and non-functional requirements,
- The second page could require students to perform simple object-oriented analysis by identifying classes, properties and operation of the classes, and relationships between classes (eg inheritance),
- The third page could be a basic design page where the students could specify the data required in the classes and the signature of each of the methods,
- The fourth page could require the students to provide the implementation of the classes by providing data and algorithms for each method,
- The fifth page could compile the separate classes, require the specification of test data for each class, and the unit testing of these classes,
- The sixth page could compile the entire application and the download of the Java applet to run within the student's web browser, and
- The seventh page could require the specification of system test data and the testing of the system on that data with the production of a test report.

The identification of classes in analysis can lead to design classes which can lead to predefined classes in implementation (feedforward engineering). With a forced iterative approach students will not have the opportunity to add classes in the implementation stage without going back to the analysis page or design page and defining the classes there. The ravenous repetition of code-compile-code-compile will be broken.

The proposed system will provide most of the important features of an IDE (including compiler, source code editor, and debugger) but with limited additional functionality (that often tends to confuse students and may be more appropriate for large software development). The student's code will be entered into Web pages, stored in the database, compiled on the server, and served back as a Java applets when they are to be run within the student's web browser.

Process Data

Recording of process data is central to the PSP approach of Humphrey. However, generally in PSP the process data is recorded manually and can require a considerable amount of effort from the developer. In the proposed system, as the process is built into the Web application it is possible to (at best) automate or (at worst) semi-automate the collection and storage of process data. Reports can be produced for students and staff . For individual students it can also compare their performance to the class as a whole.

The proposed system will also allow students and instructors to track the students' progress throughout the software development. For example, the system will record and report on the number of iterations the students undertake and the amount of work done in each stage of development. As another example, the students can be encouraged to categorise the types of errors they may in coding, and the system can then construct a "top ten errors list" and encourage the students to look for these errors in their code reviews (and, even better, to take steps to prevent these errors when coding in the future).

The system will also assist staff in the on-line marking of the students' programs. Staff will be able to run the students' programs within a browser (in the exact same way that the students do). Staff will also be able to examine all artefacts of the development process (eg analysis and design models). Finally, the system will enable the staff to record marks that can be returned to the student and collated against a class list. The system should assist considerably in the administration of programming exercises.

Object Orientation and Java

Object orientation and Java were chosen as the target paradigm and programming language because: 1) Java applets can be compiled by the server and run within the student's browser removing the need for an external commercial IDE, and 2) the current popularity and focus on teaching Java and object orientation in Universities. The focus on the development process could turn out to be as important, if not more important, than the programming language or paradigm themselves. Since development occurs within a Web environment it is also possible to provide additional instructional material where it is most needed and easily accessible.

Object Orientation offers a seamless and smooth transition from analysis, to design, and to implementation. Hence, it makes the progress of the information from page to page in the Web smooth. This proposal would still be possible in traditional paradigm but the jumps between pages may be more discontinuous (eg from data flow diagram to structure charts).

Further Issues

This is a novel proposal that takes advantage of (relatively) new three-tier web application technology and the Java run-time environment in a number of interesting ways. These include (but are not limited to) the use of a web application for teaching programming, the use of a web application for forcing a disciplined software development process, the ability to monitor the student's progress and process as they develop software, and ability to reduce the amount of plagiarism.

1 Web CASE tool for Teaching Programming

The proposal is novel in that it removes the need for a complex commercial CASE tool or IDE: saving on both the cost of the commercial tool and the time necessary to train the students in the general use of the tool. Central to this ability to remove the need for external tools is the ability of Java applets to be compiled (to Java bytecode) on the server and run on the client within a web browser (with access to a Java Virtual Machine). This is not to suggest that training in a commercial CASE tools and IDEs is not important, just that it is perhaps better left to later years. On the other hand, the proposed system does provide the (much simplified) functionality of both CASE tools and an IDE, so it is at least provides an introduction to the use of such tools.

2 Forcing a Disciplined Software Development Process

The author is not aware of any similar system that forces a disciplined software development process by not allowing the students to get caught in an endless code-compile cycle but requiring them to analyse, design, code, run, and test in an iterative cycle. The majority of the IDEs that are used in programming courses provide a simple "one-click" compile button. The temptation to code-and-compile is easily satiated in such system.

Central to the system proposed here is the fact that the Web application collects their work from each phase of development and directs them through the phases (not allowing them to jump back and forward between coding and compiling).

3 Ability to Monitor the Student's Progress and Process

The proposed system is also unique in that it allows monitoring of individual student's progress and process execution (as well as the group's progress and process) as they develop the software. As well, since all the students work is stored in a central database, it removes the need for manual collection and return of work, and assists in the automation of on-line marking. Students no longer need to submit their assignments on floppy disks (or have the ability to argue that their computer crashed at home and they lost all their code). The system can record and report on everything the students do.

4 Ability to Reduce the Amount of Plagiarism

Plagiarism is a big problem in introductory computer courses. The proposed system deters plagiarism since the entire development process is recorded and programs cannot simply be duplicated. Since the system focuses on the actual development process (not just the final code or executable application the students produce) it is much more difficult for students to plagiarise other student's work. A simple copy and paste of code will show up clearly in the record of the development process. In essence, each student has to go through all of the steps (usually a number of times) to successfully complete the software development process.

Educational Relevance

This proposed system will be of significant educational value and relevance to the tertiary (and perhaps even secondary) educational community involved in teaching programming specifically, and/or software development, in general. It will provide in-depth information (almost a complete trace) of everything students do in completing the programming exercises. It will also be of interest to those who research into the development of Web applications, as well as the way students program and the most efficacious way to teach students programming and software development.

1 Teaching Software Development.

Currently, most students hack when they program (at least in introductory level courses). There is little thought or sensible process in their development. The proposed system is highly applicable to any software development or engineering course that wishes to emphasise and teach good software development process because it focuses students thinking on analysis, design, specification, coding, and testing. In its first release, it targets object-oriented systems development, but in the future it could be extended to other development paradigms.

2 Teaching Programming Languages.

Most commercial integrated development environments (IDEs) are too complicated for teaching programming at an introductory University level. Time spent teaching students how to use the IDE is to the detriment of time spent teaching the programming language, programming concepts and a software development process. This system allows the educator to focus on the issues of importance, not the IDE. In its first release, it targets the Java programming language, but in future releases it could be extended to object-oriented and non-object-oriented programming languages.

3 Research and Pedagogical Aspects.

The development and use of the system would also allow significant investigation into Web application development methodologies and the pedagogy of programming languages and the software development process. In the latter case, the tool would allow educators to analyse student development and make instructional corrections to their process. It would seem that, for the first time, educators would have a real window into the minds of students as the learning a programming language, programming concepts, and a software development process. This will be of real value!

Summary

The current and unfortunate situation that learning to program often means learning to hack code has been identified, as well as the appropriateness for students to focus (at least as much) on the development process as the programming language or programming concepts. A number of goals and constraints have been identified within the greater challenge of teaching software development to students.

Proposed is a three-tier Web-based system that will force students to follow a disciplined software development process whilst learning a programming language and programming concepts. Development is performed completely within a (Java-enabled) Web browser that will step students iteratively through a series of Web pages incorporating process phases (with checks between phases) leading to a running program (Java applet).

The system will focus initially on object-oriented programming in the Java language. It will act as a much simplified and Web-based CASE tool and IDE, assist the students in analysis and design, compiling students programs and running them as Java applets in a Web browser. The system will also record data about students' software development process that can assist in improvement of that process (by students and staff).

Finally, the system has a number of benefits that include: the removal of the need for separate CASE tools and IDE, it can force a discipline development process, it can record and display data about the students' processes, it enables secondary research on students' approach to development, and it may help to reduce plagiarism in exercises. Most importantly though, it enables the tables to turn, and the focus to be equally on a disciplined software development process and the programming language and programming concepts.

Appendix – Implementation

This proposed system is being implemented as a Web application using WebObjects on MacOS X Server. Students' programs and process data will be stored in an FrontBase database, compiled using the JVM on MacOS X Server, and served to the student's browser to run as applets. The research is funded by a Major Research Grant from the Apple University Consortium. Also involved in the project are Alan Parkinson and Charles Zhao from the School of Information System at Curtin University of Technology.

References

Biddle, R. and T. Ewan (1998). "Teaching programming by teaching principles of reusability." *Information and Software Technology 40*(4): 203-9.

Brilliant, S. and T. R. Wiseman (1996). First programming paradigm and language dilemma. *The 1996 27th SIGCSE Technical Symposium on Computer Science Education*. Philadelphia, USA, ACM.

Humphrey, W. (1995). *A Discipline for Software Engineering*. Sydney, Addison-Wesley.

Humphrey, W. (1997). *Introduction to the Personal Software Process*. Reading, MA, Addison-Wesley.

Stroustrup, B. (1986). *The C++ Programming Language*, Addison-Wesley.

Wolz, U., S. Weisgarber, et al. (1996). Teaching introductory programming in the multi-media world. *The 1996 Symposium on Integrating Technology into Computer Science Education*. Barcelona, Spain, ACM**:** 57-9.