Using the Internet in a Computer Science Senior Projects Course

Michael Werner
Department of Computer Science, Electronics and Mechanical
Wentworth Institute of Technology
550 Huntington Avenue
Boston, MA 02115

Abstract:
Project courses in Computer Science require a student to discover a feasible software project, do independent research to see how his/her project fits in with similar and related projects done elsewhere, acquire the necessary knowledge and tools to build the project, and finally, present the project for both peer and teacher evaluation. Use of the Internet can be a powerful tool towards meeting the student’s needs.

Introduction:
This article describes the use of the Internet as a resource in a computer science senior project course. The aim of the project course is to allow the student to consolidate the knowledge and skills gained both by taking courses, and from required on-the-job coop experiences. They demonstrate their abilities in the form of an original project, which they build themselves or as part of a small team. The special nature of this kind of course necessitates a different teaching approach, one that is student-centric, in that the student bears much of the burden for deciding what knowledge should be obtained, and the mechanics for obtaining, mastering and utilizing it.

The aim of this article is to show that the Internet can be used as a key enabling technology for a student-centric course. It was successfully tried in a course given in the summer of 1998. Among the unanticipated issues that came up in the teaching of this course was a lively discussion of intellectual property rights.

Background:
Wentworth Institute of Technology is a medium-sized technical college located in Boston, Massachusetts, with concentrations in Environmental, Electronics, and Mechanical Engineering Technology, Architecture, and Computer Science (CS). For some time, the non-CS programs required students to do a senior project as part of a capstone course to demonstrate and consolidate what they had learned in the program. In 1998, this requirement was extended to include Computer Science as well.
Devising a suitable project course for CS was a challenge. Clearly the course needed to be different from the typical undergraduate course in that it would center on student initiatives. I use the term *student-centric* to describe this, as opposed to a Freshman-level programming course, which I would consider to be *teacher-centric*.

**Student-Centric vs. Teacher-Centric Courses:**

The table below clarifies some differences between student-centric and teacher-centric approaches:

<table>
<thead>
<tr>
<th></th>
<th>Teacher-Centric</th>
<th>Student-Centric</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Educational Objectives</strong></td>
<td>There is a specific body of knowledge to be mastered, as outlined in the course description. This may include learning the syntax of a specific programming language, how to analyze problems, and how to design solutions.</td>
<td>The student is expected to synthesize knowledge obtained in other courses, and apply it to a project of his/her own choosing.</td>
</tr>
<tr>
<td><strong>Syllabus</strong></td>
<td>Class meetings are tied to specific objectives.</td>
<td>Project schedule with general objectives in terms of development milestones</td>
</tr>
<tr>
<td><strong>Knowledge Transfer</strong></td>
<td>The teacher and the textbook are presumed to have the knowledge at the start, and the student not to have it. Knowledge is transferred from the teacher and text to the student. The student also benefits from experiential learning in a laboratory environment that has been carefully structured by the teacher.</td>
<td>The teacher has experience and background in designing and executing solutions to problems. The teacher and, to a lesser extent, the student have a broad background in computer science. Neither may have the specific skills required to complete the student project. The student may need to obtain these skills elsewhere.</td>
</tr>
<tr>
<td><strong>Presentation</strong></td>
<td>The teacher organizes the presentation of the material, using print or electronic media, and classroom lectures.</td>
<td>The student organizes the presentation. The student may present materials in person, or remotely to the teacher, and also to others in the class for peer review. Presentations include summaries of research results, lists of requirements, design documents, progress reports, and project implementations</td>
</tr>
</tbody>
</table>
Evaluation

<table>
<thead>
<tr>
<th>Grade Method</th>
<th>Specific Objectives</th>
<th>Project Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teacher</td>
<td>Testing attainment</td>
<td>Research effort, student presentations, degree of project implementation</td>
</tr>
<tr>
<td>Student</td>
<td>Effectiveness in presenting material</td>
<td>Effectiveness of presentations, extent to which student was helped</td>
</tr>
</tbody>
</table>

Pilgrim and Leung\(^1\) make the salient point: *History reminds us that instructional systems should be developed with regard to a strong theoretical and philosophical foundation that is suitable for the technology utilized.* In this case, it is the philosophy of student-centric education, which drives the search for an appropriate technology. This search led almost immediately to the Internet. Use of Internet resources provides students with the means to obtain the needed knowledge, to organize and present their work for peer review and evaluation by the professor, and finally to publish their results.

Student-centric education can also take advantage of research on Problem-Based Learning\(^2\) (PBL), a technique which has been widely adopted in medical schools. It is based on Bruner’s\(^3\) psychological theory of discovery (inquiry) learning. Russel Kenley\(^4\), an architecture professor, analyzes PBL in terms of Wood’s\(^5\) three categories of teaching methods, namely: teacher-based, text or media-based, and problem-based. Kenley did experiments using PBL in classes in project organization, and construction management. He found it critical that students be able to access relevant material in a PBL course. Access to current research is one of the great strengths of the world wide web, particularly in areas such as computer science, where practitioners are often web-literate, and willing to share their results by freely posting them.

Using the Internet in Teaching a Student-Centric Projects Course.

The availability of Internet resources proved to be the key to offering a successful Computer Science Projects course. The world wide web was used in the following ways:

1) **For presentation using student web sites.**

   Each student presented his/her work on an individual web site. A course requirement was for each student to create a personal web site, either on a service provided free by Wentworth Institute, or on an Internet service provider of their own choosing. Students were required to learn enough HTML to present their projects. Their project pages needed to have a proposal, requirements study, and supporting design documents. In addition, students were encouraged to provide down-loadable executable code so that visitors to the page could try out the projects, and down-loadable source code so that visitors could see how they were programmed, and possibly make enhancements.
Some students exceeded the minimum presentation requirement, producing sophisticated web sites. As far as placing source and executable code on the sites, the students split into three groups. Students in the first group readily complied with the request in the spirit of fully sharing their accomplishments. Those in the second group saw some commercial potential in their work, and were concerned about others stealing their ideas before they were able to properly copyright them. Students in the third group had a different concern, namely that they themselves might have inadvertently violated a copyright and would be opening themselves up to a legal action if they displayed the details of their work.

2) **For knowledge acquisition:**

As part of their project proposal, each student was required to do a world wide web search to find similar projects. In particular, students developing software were required to search some large shareware sites, including www.gamelan.com, and www.jumbo.com. They also did keyword searches using search engines such as Alta Vista. Having found similar projects, the students then needed to make the case that their project was differentiated from them, say by offering enhancements. Students also provided links to the similar projects from their own project pages.

3) **Development of web tools**.

Several student projects involved developing Internet tools including:

a. **A children’s web browser.** A two-person team did some research into how to make a web browser that would be easy for children to use. Among their findings was the need for large buttons, less confusing error messages, and more visible activity on the screen during long down-loads. One issue with error messages was the need to reassure children that they had not done anything wrong, if for example, a URL was not found. After identifying a number of desirable features, a working browser was then built incorporating them.

b. **A bulletin board manager.** A software utility was developed to facilitate the day-to-day operation of a web-based bulletin board.

c. **A user manager for an Internet service provider (ISP).** This utility, programmed in Java, allows a manager of an ISP to easily enroll or drop users, and to monitor usage.

d. **An FTP client written in Java.** The novelty of this client is that it allows resumption of interrupted file transfers. It also has both a command interface and a graphical user interface. The student who built it originally searched the Internet until he found a command-based FTP client that had been written as a student project at another University. He intended to add a GUI interface to it as his project, but in the course of doing so, he discovered that the original software
was unreliable, and he ended up rewriting that as well. In the end, he published his program as freeware on the web so that others can benefit.

4) **Classroom progress reports and peer reviews.**

Students met as a class for one hour each week. This was in addition to a weekly individual meeting with the instructor. The class meeting was conducted in a media equipped room. After the first few classes, the instructor took a back seat, and turned the time over to the students. Typically students would show design documents, and other materials, which they had posted on their individual web sites. There was often a lively discussion of tools, techniques, and user interface issues. This mode of presentation meant that little time was lost in setting up between presentations. About half the class presented each week.

5) **A course page for disseminating information.**

The professor set up a page for the course, including links to each of the student project pages. This page also had the syllabus, class notes, and other helpful links.

6) **For evaluation and grading.**

Grades were determined in part by the effectiveness of the presentation of the project on the student’s web page.

**Providing Facilities:**

To complete the requirements of the Projects course, students needed access to adequate computer facilities:

1) **Ready availability of fast connections to the worldwide web.** Fortunately, there is good access from Wentworth, both from computer laboratories, and from dormitories. Fast connections are crucial to making this approach work.

2) **Use of a server to host student web pages.** While students at Wentworth were provided space on a public web server run by the institute, they were limited in file space, and were not allowed to install servers, or CGI scripts. Many students used outside resources which they had access to, instead.

3) **Use of software suitable for preparing HTML pages.** No such software was provided. Instead, students downloaded freeware, or preliminary versions of commercial programs. While this put them on the cutting edge, it also meant that their pages required numerous plug-ins, and often were not accessible from older browsers.

4) **Availability of training in use of HTML software.** There was no formal training. This lack of training did not prove to be a major handicap. Some students were already proficient in preparing web pages, and were more than willing to help others. This
assistance is in keeping with a student-centric approach, and worked well in practice. However, it is necessary to keep in mind that these were Computer Science students. In other disciplines some training may be advisable.

How Successful Was The Course?

Student evaluations.

Out of 18 students polled in an anonymous evaluation, all agreed including 7 who strongly agreed to: This course challenges my abilities. Similarly, all agreed and 7 strongly agreed to: Overall this course was a positive experience. Since this was a first time offering, there was no previous data with which to compare. There were some comments recommending that the course be expanded to 2 semesters, so as to allow more time for developing substantial projects.

Professor evaluation.

Most students were able to add substantial projects to their portfolios. Furthermore, these were in a form that could readily be displayed to prospective employers or other interested parties. Several students intended to continue working on their projects after graduation, either by seeking related employment, or by pursuing them as a hobby. Nearly all students were able to consolidate and reinforce skills they had learned in prior courses. A few students were reluctant to learn HTML at first, but in the end all had achieved at least a minimal mastery of the language. Some went beyond this to put together well-designed and attractive web sites.

Intellectual Property Rights Issues:

The issue of intellectual property rights was not included in the original syllabus for the course, but came up for major discussion anyway. Specific concerns were:

(1) Could students build computer games based on copyrighted board games?

(2) Could student projects include scanned images, or images obtained from the net?

(3) When a student project was an enhancement to someone else’s project found on the web, how and to what extent should the original project be acknowledged?

(4) How could students protect their own ability to further develop their projects, and possibly exploit them commercially, while at the same time exposing them to view on the web?

(5) Could students include various software plug-ins in the installation kits for their own programs?
Many computer science programs of study now include a required course in ethics, which would be a good place to address intellectual property rights issues. It is recommended that such courses consider cases from both an ethical and a legal point of view, since the two do not always coincide. The Wentworth students in the study had not taken an ethics course, requiring that class time in the project course be spent on these issues.

The discussion was lively. Many students began seeing their own work as being worthy of copyright protection. This gave them a new perspective on the ethics of intellectual property rights. Others became concerned that they may have violated copyright laws, opening themselves to lawsuits. They began to read licensing agreements on software they had downloaded from the Internet. However, neither the students nor the professor were particularly knowledgeable about these issues. Fortunately, the Internet itself proved to be a good resource for obtaining the needed information.

There are some extremely good on-line sources for information about intellectual property rights issues. A good starting point is BizTech, which has pointers to a number of other sources. The U.S. Copyright Office provides information on copyright registration, circulars, and related copyright topics. A good student perspective by Matt Rosenberg is available from the ACM magazine Crossroads.

Not all students were interested in protecting their work. A few were more than willing to make all of it available, including design documents and source code, in the spirit of world-wide sharing. One student, in addition to his programming project, did a report to the class on the Open Software Foundation, an organization dedicated to providing unrestricted access to software without charge. The report gave a background on the OSF’s copyleft licenses and the role of the OSF in developing the alternative operating system, LINUX.

**Conclusion:**

Use of the Internet proved to be a key element in making a student-centric approach to a Projects course work. It allowed the student to do required research, present designs for peer review, and display work as part of a portfolio. It also was useful to the professor for classroom management and communication needs. While the results are based largely on experience in a single course, they were encouraging enough to merit application in other courses.

The experience clearly demonstrated that students are motivated and able to acquire the knowledge needed to complete projects from the Internet, rather than from the professor. The professor in turn needs to function as a coordinator, facilitator, and evaluator, and is also responsible for providing the minimum infrastructure. Finally, it should be noted that students become more self-reliant and confident as a result of their usage of the Internet.

---

MICHAEL WERNER
Michael Werner, 

Michael Werner, wernerm@acad.wit.edu is a Professor of Computer Science at Wentworth Institute of Technology in Boston. His main interests are in bringing objected-oriented analysis, design and implementation techniques into undergraduate software engineering courses. He is currently completing a Ph.D dissertation at Northeastern University entitled Facilitating Schema Evolution With Automatic Program Transformations.