A Teaching Portfolio

Craig E. Wills

Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609

October 23, 1998

1 Introduction

The primary goal of this teaching portfolio is to set forth my philosophy on teaching and describe specific practices I use in the classroom. It also describes educational and curriculum projects I have been involved with. This portfolio is intended to be dynamic and incorporate new ideas and approaches. Some of this work has led to sponsored research, publications and presentations, which is referenced as appropriate. The portfolio includes results on the effectiveness of my teaching based on student feedback and advising on projects based on project outcomes. The portfolio concludes with a statement of my future goals and new directions for investigation.

2 Teaching Responsibilities

As background for this portfolio, I have taught courses at WPI as a full-time faculty member, at Victoria University of New Zealand while on sabbatical and at Purdue University as a graduate student. At WPI, I have taught a range of courses from a large, introductory level programming course to an advanced graduate course.

The undergraduate courses I have taught are: CS2005 (Techniques of Programming), which is a large (100-200 students), introductory course concentrating on data structures and advanced programming; CS3013 (Operating Systems I), which is a junior-level course covering fundamentals of operating systems; CS3041 (Human-Computer Interaction), which is a junior-level course on principles of user interfaces; and CS4514 (Computer Networks), which is a junior/senior-level course covering all aspects of computer networks.

The graduate courses I have taught are: CS502 (Operating Systems), which is a core course covering a broad range of operating system topics; CS513 (Computer Networks), which covers fundamentals of computer networks; and CS535 (Advanced Operating Systems), which covers a broad range of advanced and distributed operating system topics.

3 Teaching Philosophy and Goals

I believe each one of us has a particular combination of interests and abilities that make us best suited for a particular job. The closer we can come to this job, the closer we come to reaching our full potential and attaining a sense of satisfaction with our lives. As students grow they begin this search for what they want to do with their life. I believe it is our responsibility as teachers to provide students with opportunities for them to find and demonstrate their abilities. Student potential is only realized if it is nurtured. This goal drives me as a teacher to provide a positive environment for each of my students to grow and perhaps find a direction of what interests them. In this way, I often think of my role as a coach in the

classroom where I am trying to bring out the best of what each student has to offer in a particular course or project.

The environment that I believe best accomplishes this goal, and the one that I try to foster, is challenging and supportive. Without challenge students do not extend themselves, do not realize potential and become disinterested. Without support students feel their effort is unappreciated, the expectations unrealistic and become discouraged. I try to create a classroom environment that encourages dialogue between me and the students making it easy for them to ask questions and make comments. This approach requires and leads to mutual respect. In conjunction, I believe the best assignments have open-ended aspects leaving room for students to make the assignment their own. When students take ownership, they take responsibility and do work they are proud to show me and their peers.

I believe two other aspects are crucial to effective teaching. The first aspect is the need for flexibility in the techniques used for teaching a course. It is important to match the approach with type of class and nature of the students. The second aspect is that good organization is essential. Overall, the students need and want to feel that the course is well managed and that the instructor has clear objectives. This situation breeds confidence in students and allows them to concentrate on the course material and not worry about the course organization.

4 Projects and Innovations

This section highlights projects and innovations that I have been involved with both in specific courses and in curriculum development. References to sponsored work, publications and presentations are included as appropriate.

4.1 Introductory Computer Science Curriculum

I helped draft the description for the course CS2005 in Spring, 1991. This course is now the gateway course for majors and non-majors seeking to take upper-level Computer Science courses. It primarily covers advanced programming and data structures.

I was one of four members of a "brown bag lunch" group (along with Profs. D. Finkel, M. Gennert and M. Ward) that drafted proposed curriculum changes for the Freshman/Sophomore Computer Science course sequence in the Fall, 1991. These changes were ultimately adopted by the department and form the introductory course sequence for our major. This involvement also led me to be an invited participant in a three-year, NSF-sponsored Workshop on Breadth and Laboratories in the Introductory Computer Science Curriculum. The workshop was sponsored by faculty from Bowdoin College, Clemson University and the University of Connecticut. Each summer from 1992 to 1994 I was one of thirty participants from schools across the country that gathered at one of the sponsoring schools and dis-

cussed innovations in the introductory computer science curriculum. This work and my efforts on the use of peer learning has led to being known in the computer science education community.

4.2 Peer Learning

One innovation I have introduced is an emphasis on the use of peer learning, particularly in the course CS2005. Thanks to a grant from the Davis Foundation through the Center for Curricular Innovation and Educational Development on campus [15], I was able to introduce reforms to have students work in groups on programming projects. These student groups were supplemented with upper-level undergraduates (peer learning assistants) to facilitate interaction within each student group. My motivations in using group projects were to continue to provide a support structure for students in a large course and have them take more responsibility for their own learning.

This work on with peer learning has not only been successful in the classroom, but has led to a number of publications and workshops with colleagues in the department, on campus and at other institutions across the country [18, 24, 6, 23, 17, 28, 11].

As an outgrowth of this work, I received National Science Foundation funding for the "Application of Peer Learning to the Introductory Computer Science Curriculum" [13]. This project began with a workshop in the Summer, 1996 with a follow-up workshop in the Summer, 1997. The project web page can be found at `http://www.cs.wpi.edu/~peercs`. Papers and presentations have resulted from this work [14, 22].

4.3 Integration of Research and Curriculum

This work has focused on integrating my research interests in distributed computing, networking and user interfaces into the undergraduate curriculum. A focus of the work has been the creation of "The Webware, Interfaces and Networking Experimental Laboratory" (WINE Lab) through support of the National Science Foundation [16]. This lab allows students to not only understand the important concepts of each domain, but to better understand the effects of the interaction between the domains. Papers and presentations have resulted from the work [19, 5]. The lab Web page is available at `http://www.cs.wpi.edu/Resources/WINE/`.

5 Teaching Practices

This section provides details on specific practices I use, which result from the philosophy described in Section 3. Many of these efforts are a result of trying to improve my teaching.

5.1 Classroom Approach

One approach I use for encouraging interaction with the students and to create a relaxed classroom atmosphere is to begin each class by sitting on a table in the front of the room. I use this forum to make general announcements and answer any questions before we get into the material for that class.

My general classroom approach is to not use prepared overhead slides for introducing material. Rather, I typically work from prepared notes and write important material on the board. I feel this approach works well because it helps provide better pacing for me to present material and for students to take notes. Interspersed with my presentation, I ask for student comments and questions.

I do supplement my notes with prepared slides for diagrams or program code that is not practical to write on the board. In these cases I try not to put any material on the overhead that I expect students will want to copy into their notes. I avoid this problem by using examples from the text and making sure that any program code is available online.

5.2 Use of Technology

I routinely use electronic mail in courses as well as make my course notes and old exams available on-line to the students through a course Web page. I also introduced the department to the program *turnin*. This program allows students to turn in programming assignments electronically rather than via printouts. This approach not only saves paper, but makes grading of these assignments more thorough. The use of *turnin* has been adopted by virtually all courses in the Computer Science Department.

I developed software for the management of student course registration information. The software is used to form student groups for projects and collect student evaluations on the group projects. The software was used for the first time in CS2005 in B Term 1993 and has been used by many other courses on campus.

5.3 Hands-On Project Experience

Good projects need to be assigned to help students learn concepts and these projects need to be integrated with the course material. In the introductory and systems courses I have taught, programming projects are crucial in helping students gain first-hand experience with applying concepts and techniques from the course. I often build try to build the course around the set of projects used in a course. Due to the condensed nature of our undergraduate term, my approach is to hand out the first project on the first day of the class to help gain momentum for the class. My experience with projects used in the undergraduate Computer Networks course has been published [26].

5.4 Student Ownership

Another topic related to student projects is to have projects that allow students to take ownership of the project and to share their work with their peers. Not all of my assignments have this characteristic, but I have found that such projects are both popular with the students and motivate them to do better work. One example using this approach is a client/server project in Computer Networks where students create their own network service, which is advertised and can be used by the rest of the class. Example services include checkers, other games and a weather service. Another example is in Human-Computer Interaction where students create an interface to an application and then evaluate each other's work. This approach is useful not only for the person being evaluated, but the person doing the evaluation. At the graduate level, I let students take ownership of a piece of the Advanced Operating Systems course by allowing them to select their own project and then have each student make a presentation of their work.

5.5 Approach to Grading

My approach to assigning and grading student work, particularly at the undergraduate level, has changed during my time at WPI. I have moved to a "criterion-influenced" grading policy based on discussions with Jim Groccia, former head of the Center for Curricular Innovation and Educational Development when I first thought about using group projects. I was also influenced by attending teaching seminars given by Professor Jack Boyd of Mechanical Engineering,

The basic idea of my approach is that meeting the basic expectations for a course results in a grade of B with opportunities for students to demonstrate additional understanding and insight for a higher grade. The following quote is extracted from a typical course syllabus on how I present this approach to the students.

> Grading policy for each project and homework will be provided at the time of the assignment. In general, each assignment will have a basic objective for the majority of the assignment points and an extended objective for demonstrating additional work and understanding.
>
> Final grades will reflect the extent to which you have demonstrated understanding of the material, and completed the assigned projects. The base level grade will be a "B" which indicates that the basic objectives on assignments and exams have been met. A grade of "A" will indicate significant achievement beyond the basic objectives and a grade of "C" will indicate not all basic objectives were met, but work was satisfactory for credit.

With this approach, a student meeting the basic objective for an assignment

typically receives about 85% of the total assignment grade. The level of difficulty for the remaining 15% is often much harder.

I also use this same approach for testing students on exams. For exams, I create a "list of expectations" for each exam on what I expect the students to know based on the material we have talked about. This list both focuses students on what material is most important and provides a detailed list of what instructors in follow-up courses can expect students to know. For the exam itself, I typically base about 85% of the exam on the list of expectations. The remaining 15% requires students to extend the material they are expected to know in ways that we have not discussed. The following are typical expectations from the CS3013 Operating Systems course.

> Know the different types of operating system structure. Know why client-server structures are increasingly used.
>
> Understand how the operating system gives the illusion of executing multiple processes at the same time.
>
> Understand the possible states of a process and the transitions between states.

My overall approach to grading has some elements of mastery learning in testing them on a set of basic criterion. However it is not strictly mastery learning because I do not have students repeat an assignment or exam if they do poorly. I consider the approach more "criterion-influenced" than "criterion-based" because I don't set strict strict standards on what level students must attain nor do I grade the basic objective portions separately.

5.6 Student-Written Exams

I have had trouble with courses where the range of covered material is broad and does not easily lend itself to traditional examination means. Two such courses I have taught are CS3041 (HCI) and CS535 (Advanced Operating Systems). I have experimented with traditional instructor-prepared final exams, which are difficult to create and difficult for the students to study for. I have experimented with not using any exams in such courses, but I have felt that such courses did not have good closure of the material.

The approach that I have adopted with such courses is to let the students write the final exam. The methodology is that students each submit a list of 3-4 questions near the end of the course. The lists are then merged and pruned to avoid duplicates or obviously bad questions. As part of the process, I add as many questions as needed to round out the coverage of course material. I then give the entire list of questions back to the students and inform them that the final exam will consist of a subset of this list. Students are not allowed to bring notes with them to the exam as I had a bad experience when I allowed students to do so. In

general, I have found the exam grades a little higher than for traditional exams, but the exam does accomplish the purpose of having students review the important concepts of the course and in that way providing closure on the course contents.

5.7 Peer Learning in the Classroom

Motivated by my work on group projects in CS2005, I more generally use in all classes an activity that I call a "group quiz." In this in-class activity students group themselves based on seating proximity. The quiz may require the students to consider different design possibilities or apply concepts learned in class to a different problem. While the students work and discuss, I circulate through the classroom answering questions and facilitating discussion. Particularly in a large lecture, this technique creates different course dynamics by me being amongst the students instead of standing in front. After the groups have worked on the exercise for some amount of the meeting time, the class spends time discussing problems encountered.

Groups are responsible for making sure each member understands the work that is done. One member records the group membership and results, turning in the work at the end of the class. This approach allows me to easily review all, or a sampling, of the work submitted, as well as give credit to all students participating in the activity. I believe correctness of the results is not as important as promotion of the active learning process in this activity. I typically test understanding by giving a similar exercise on a subsequent exam.

6 Student Response

6.1 Course Evaluations

Student evaluations across the different types of courses I teach have been strong and consistent. The evaluation results indicate an average of 95% of the students agree or strongly agree with the statements of Part I of the standard WPI evaluation form.

6.1.1 Undergraduate Courses

Undergraduate courses at WPI are taught on a 7-week term schedule. Each course typically consists of four class hours per week with students taking three such courses each term. Terms A and B are taught in the time period of a traditional Fall semester and terms C and D in the time period of a Spring semester. The evaluation results given below are from Part I of the standard WPI evaluation form. They indicate the percentage of students who agreed or strongly agreed with fourteen specific perceptions about the instructor of the course.

Course	Title	Term	Enrolled Students	A/SA on Part I Student Evals
CS2005	Techniques of Programming	B, 1996	187	94%
CS3013	Operating Systems I	A, 1996	70	99%
CS3013	Operating Systems I	C, 1996	47	96%
CS3013	Operating Systems I	A, 1995	90	98%
CS4514	Computer Networks	C, 1995	72	98%
CS2005	Techniques of Programming	B, 1994	130	92%
CS2005	Techniques of Programming	B, 1993	128	95%
CS3013	Operating Systems I	A, 1993	57	95%
CS3041	Human-Computer Interaction	B, 1992	47	89%
CS3013	Operating Systems I	A, 1992	63	93%
CS3041	Human-Computer Interaction	B, 1991	42	95%
CS3013	Operating Systems I	A, 1991	49	96%
CS4514	Computer Networks	C, 1991	53	97%
CS3041	Human-Computer Interaction	B, 1990	42	95%

6.1.2 Graduate Courses

Graduate courses at WPI are taught on a traditional semester basis. Course evaluations are the same used for undergraduate courses.

Course	Title	Semester	Enrolled Students	A/SA on Part I Student Evals
CS535	Advanced Operating Systems	Spring, 1997	11	97%
CS590	Graduate Seminar	Fall, 1996	5	100%
CS513	Computer Networks	Spring, 1996	35	97%
CS535	Advanced Operating Systems	Fall, 1994	14	95%
CS513	Computer Networks	Spring, 1994	31	97%
CS535	Advanced Operating Systems	Fall, 1992	19	96%
CS502	Operating Systems	Fall, 1991	42	95%
CS502	Operating Systems	Fall, 1990	12	98%

6.2 Student Comments

To better understand the reasons for the numerical results from the student evaluations, I have analyzed the written comments on the back of the standard WPI course evaluation for some courses. I encourage students to write comments and in some cases have asked students to address a particular aspect of the class in making comments. Analysis of written comments is difficult and subjective, but I have grouped similar types of positive and negative responses in the following tables. Each entry shows the percentage of evaluations

6.2.1 Positive Comments

Course	3041	4514	3013	2005
Term	B91	C95	A95	B96
Number of Evaluations	35	51	61	126
Number of Written Responses	35	43	49	86
Positive Comment on:	percentages			
lecture style/communication	9	12	10	14
course organization	3	7	0	3
web page/online notes	0	5	31	3
projects/assignments	29	26	29	8
material/subject	0	14	8	2
good/best professor	6	12	22	14
good/best course	6	9	2	1
no exam	46	-	-	-
group projects	-	-	-	10

6.2.2 Negative Comments

Course	3041	4514	3013	2005
Term	B91	C95	A95	B96
Number of Evaluations	35	51	61	126
Number of Written Responses	35	43	49	86
Negative Comment on:	percentages			
lecture content/style	0	0	6	3
project writeup/relevance	3	0	0	1
more office hours	0	0	2	2
overall course	3	0	0	0
group projects	-	-	-	15

6.2.3 Analysis of Comments

The available written comments reflect the numerical evaluations in being strongly positive. Many of the written comments are not explicitly positive or negative or are difficult to characterize so the relative percentages are low.

Lecture style and course organization receive many positive comments and a few negative ones. I enjoyed one comment in CS2005/B96 characterizing my attitude in answering questions at the beginning of class as a "fireside chat." The availability of a course Web page and online notes received a lot of positive comments in CS3013/A95 because I explicitly asked students to comment. Many students liked the projects and assignments in the classes, which corresponds to a focus of my course preparation.

Looking at more general comments, both Computer Networks and Operating Systems as a subject are of interest to students. There are some comments explicitly indicating that I did somewhere from a good to "the best" job in teaching the course. There was a nice comment from one of the students in CS2005/B96 indicating a change of major to computer science because of the course. In course-specific comments, the lack of an exam in CS3041/B91 was highly popular amongst the students (although I felt it did not provide good closure and switched to a "student written" final exam). The group project model in CS2005/B96 elicited both positive and negative comments from the students. Fewer comments on the group projects were received because I had previously surveyed students on the group projects.

6.3 Teaching Honors/Awards

I have received two honors and awards related to my teaching efforts:

1. Selected as an honorary member of the WPI Upsilon Pi Epsilon (UPE) computer science honor society. Fall 1995. Quoting from the invitation letter from Scott Salvidio, chapter president, "Your election as an honorary member of Upsilon Pi Epsilon signifies the high regard we have for your work in the field of computer science as a faculty member of WPI. The level of dedication and hard work you exhibit while educating the members of the computer science student body is second to none. Considering the nature of your position in the computer science department we feel that your contributions to the WPI CS community are all the more deserving of special recognition."

2. Best Teaching Assistant Award in Department of Computer Science, Purdue University. 1984.

6.4 Projects Advised

WPI has a project-based undergraduate curriculum, requiring each student to complete a Humanities Sufficiency, an Interactive Qualifying Project (IQP) and a Major Qualifying Project (MQP). The IQP challenges students to examine how science or technology interacts with societal structures and values. Credit for the project is the equivalent of three courses at WPI. The MQP is done in the major field of study (typically as a senior) and is a capstone experience for demonstrating application of skills, methods and knowledge of the discipline. Credit for the project is the equivalent of three courses at WPI.

I have advised 25 MQPs in the areas of networking, operating systems and user interfaces. Many of the projects have led to high quality and original work as evidenced by multiple awards for best MQP of the year honors within the

department [1, 9, 8, 4, 12] and many published papers from MQP work [7, 2, 27, 25, 20, 21, 3, 10]. I have advised six IQPs, principally dealing with the impact of networking and the Internet on society.

7 Future Teaching Goals

My long term teaching goals are to continue to be an effective educator while seeking innovative methods to help students learn. As part of this work I plan to continue to be active in the computer science education community and contribute results of my work to the larger debate.

In thinking about issues and ideas for teaching, I believe there are three fundamental questions that must be asked about a new approach.

1. Is it going to be pedagogically effective?
2. Is it innovative?
3. Is it maintainable over the long-term?

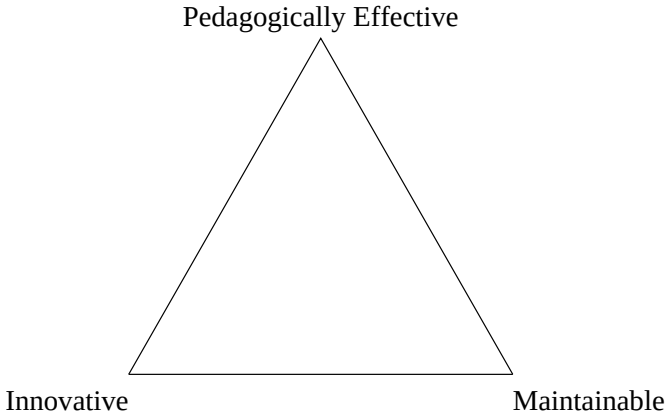

Figure 1: Trade-offs in Adopting New Teaching Approaches

These three questions lead to potential trade-offs to consider, illustrated by the three points of the triangle in Figure 1. Innovative approaches are important to consider, particularly if one hopes to attract external interest and funding. However, if an approach is not going to be pedagogically sound as an effective learning environment for students then it is not good. Similarly, an approach may be both innovative and pedagogically sound, but if it is so costly measured in money or faculty time that it is not going to be maintainable over the long term. Ideally new approaches and techniques find a middle ground amongst each of these three considerations. Keeping in mind these considerations, I have an ongoing list of approaches that I plan to explore.

7.1 Student Portfolios

I have thought about the idea that students maintain a portfolio of work throughout a course, which would allow students to demonstrate what they have learned and understand in the course. The portfolio would include assignments and exams that I provide for students to demonstrate understanding. In addition, students could provide other evidence of learning such as other projects they do, work experience or readings and reviews. A Student would use his/her portfolio as evidence in submitting a final written statement indicating the course grade that the student believes s/he should receive. I would use the student's evaluation and portfolio in deciding on the student's grade.

This approach to assessment is appealing because it combines traditional measures such as projects and exams with other measures that students may feel are more accurate portrayals of their understanding of the material.

7.2 Use of Technology in the Classroom

While the subject of computer science is inherently about technology and I routinely use electronic mail and a course web page as part of my teaching, I use a "low tech" approach in the classroom. I do not use prepared slides with a package such as PowerPoint nor I have ever used an online demonstration on a computer to illustrate a concept. While my approach in the classroom is intended to create what I believe is a good environment for learning and interaction, I have wondered how and if I should include a computer in the classroom. I have also wondered about the impact of a classroom where every student has a computer on his/her desk and how I would adapt my teaching style to use this technology.

7.3 One Minute Papers

In a one-minute paper the instructor asks students two or three questions at the end of a class about what students did and did not understand in a class. Students write a little bit and turn in their answers anonymously as they leave. This technique provides a lot of meaningful feedback to instructors, but is one that I have used only a little. It is a technique that I believe I should use more.

7.4 Use of Lecture Time

As part of my work on peer learning and thinking how to use it, I have come to view "lecture time" a bit differently. Rather than view it as the time in which I should lecture, I believe it should be viewed as the time in which all students have reserved time for the course on their schedule. I think a challenge is to understand how to make use of those reserved hours each week as a catalyst for learning. While I do not believe that lecturing should go away, I am more conscious that this time

does not have to be used in a fixed manner. I am interested in exploring how class time should be used given this perspective.

8 Teaching Portfolio as a Dynamic Document

It is my intention that this teaching portfolio is dynamic and continue to evolve, particularly the sections on techniques I have used, ideas to try and student response. In addition, the following items are currently missing and probably should be included:

- examples of course handouts,
- examples of student work, and
- peer reviews of my teaching.

References

[1] Brian Brennan and Chris Brennan. Distributed computing through the web. Technical Report MQP-CEW-9603, Worcester Polytechnic Institute, Fall 1996. Co-advised with D. Finkel. Selected as the top Computer Science MQP in Spring, 1997.

[2] Brian Brennan, Chris Brennan, David Finkel, and Craig E. Wills. Java-based load distribution on the world wide web. In *Proceedings of the International Network Conference*, pages 9–14, Plymouth, United Kingdom, July 1998. `http://www.cs.wpi.edu/~cew/papers/inc98/inc98.html`.

[3] C. Council, E. Felton, C. Johnson, R. Mason, R. Rubinstein, and C. Wills. A virtual reality world builder. In *Proceedings of CONVERGENCE: The Fifth Biennial Symposium on the Arts and Technology*, pages 115–121, New London, CT, March 1995.

[4] Chad Council, Erik Felton, Craig Johnson, and Bob Mason. Virtual reality world builder. Technical Report MQP-RSR-9304, Worcester Polytechnic Institute, Spring 1994. Co-advised with R. Rubinstein. Selected by the Computer Science Department for presentation to parents and prospective Freshmen. Selected as a co-winner of the top Computer Science MQP in Spring, 1994. Listed with available virtual reality software in Pix-Elation, Number 11, November 1994, a publication of VRASP and Virtual Reality World magazine, a product of Meckler Media.

[5] Isabel Cruz, David Finkel, Robert E. Kinicki, and Craig E. Wills. Experiences with the webware, interfaces and networking experimental laboratory. In *Proceedings of the ACM SIGCSE Conference*, New Orleans, LA, 1999. Submitted for publication. September 1998.

[6] David Finkel and Craig E. Wills. Computer supported peer learning in an introductory computer science course. In *ACM SIGCSE/SIGCUE Conference on Integrating Technology into Computer Science Education*, pages 55–56, Barcelona, Spain, June 1996.

[7] David Finkel, Craig E. Wills, Brian Brennan, and Chris Brennan. Distriblets: Java-based distributed computing on the web. *Internet Research*. Submitted for publication, August, 1998.

[8] Chris Kmiec and Greg Snyder. Persistent information retrieval on the internet. Technical Report MQP-CEW-9302, Worcester Polytechnic Institute, Spring 1994. Selected by the Computer Science Department for presentation to parents and prospective Freshmen. Honorable mention as the top Computer Science MQP in Spring, 1994. Described in the WPI Journal article "Net Assets" by Michael W. Dorsey, Summer 1994.

[9] Patrick R. McManus and Scott Salvidio. Explaining a network. Technical Report MQP-CEW-9505, Worcester Polytechnic Institute, Fall 1996. Selected as a co-winner of the top Computer Science MQP in Spring, 1996. Resulted in paper "A WPI Undergraduate Project Explaining a Network" for publication in the Northeastern Small College Computing Conference, April 1996.

[10] Patrick R. McManus and Scott Salvidio. Explaining a network. In *Proceedings of the New England Small College Computer Conference*, West Hartford, CT, April 1996.

[11] Judith Miller, Bill Farr, and Craig E. Wills. Task design for cooperative learning in math and science courses: Covering the material and learning it better. In *14th Annual Lilly Conference on College Teaching*, Oxford, OH, October 1994. Accepted workshop.

[12] Greg Shapiro and Bevan Wang. Quantum leap: Tools for distributed computing. Technical Report MQP-CEW-9103, Worcester Polytechnic Institute, Spring 1992. Selected by the Computer Science Department for presentation to parents and prospective Freshmen. Selected as the top Computer Science MQP in Spring, 1992.

[13] C.E. Wills. Application of peer learning to the introductory computer science curriculum, June, 1996–May, 1998. National Science Foundation Undergraduate Faculty Enhancement Grant DUE955406. $56,521.

[14] C.E. Wills, D. Cordes, D. Deremer, B.J. Klein, R.A. McCauley, and L. Null. Application of peer learning to the introductory computer science curriculum. In *Proceedings of the ACM SIGCSE Conference*, pages 373–374, San Jose, CA, March 1997. Panel presentation.

[15] C.E. Wills, D. Finkel, M. Gennert, and M. Ward. Community learning in an introductory computer science course, June, 1993–May, 1994. WPI Davis Educational Foundation Program. $20,880.

[16] C.E. Wills, D. Finkel, G.T. Heineman, R.E. Kinicki, and M.O. Ward. The webware, interfaces and networking experimental laboratory, June, 1997–May, 1999. National Science Foundation Instrumentation and Laboratory Improvement Grant DUE9751132. $44,256.

[17] Craig E. Wills. Application of peer learning to the introductory computer science curriculum, April 1997. Invited presentation to the Computer and Information Science Department, Temple University. Philadelphia, PA.

[18] Craig E. Wills. Group-based software engineering in an introductory computer science course. In *Proceedings of the Software Engineering: Education & Practice Conference*, Dunedin, New Zealand, January 1998. IEEE Computer Society Press.
`http://www.cs.wpi.edu/~cew/papers/seep98.ps.gz`.

[19] Craig E. Wills, David C. Brown, Isabel Cruz, David Finkel, George Heineman, Robert E. Kinicki, and Matthew O. Ward. The webware, interfaces and networking experimental laboratory, February 1998. Invited presentation of NSF-sponsored projects at SIGCSE'98. Atlanta, GA.

[20] Craig E. Wills, Kirstin Cadwell, and William Marrs. Customization in a unix computing environment. In *Proceedings of the 7th USENIX System Administration Conference*, pages 43–49, Monterey, CA, November 1993.

[21] Craig E. Wills, Kirstin Cadwell, and William Marrs. Sharing customization in a campus computing environment. In *HCI International '93*, pages 105–115, Orlando, FL, August 1993.

[22] Craig E. Wills, Dorothy Deremer, Renee A. McCauley, and Linda Null. Studying the use of peer learning in the introductory computer science curriculum. *Computer Science Education*. Submitted for publication September, 1998.

[23] Craig E. Wills and David Finkel. Experience with peer learning in an introductory computer science course. *Computer Science Education*, 5(2):165–187, 1994.

[24] Craig E. Wills and David Finkel. Study of a group project model in computer science. In *Proceedings of the ASEE/IEEE Frontiers in Education Conference*, Pittsburgh, PA, November 1997.
`http://www.cs.wpi.edu/~cew/papers/fie97.ps.gz`.

[25] Craig E. Wills, Joachim Heck, and Ramin Taraz. Visualization of a user's information space. In *Proceedings of the Computer Science Conference*, pages 94–101, Phoenix, AZ, March 1994. ACM.

[26] Craig E. Wills, Robert E. Kinicki, and David Finkel. Networking projects in the undergraduate curriculum. *Journal of Computing in Small Colleges*, 11(4):238–245, March 1996. Based on a presentation at the First Annual Northeastern Small College Computing Conference. West Hartford, CT. April 1996.

[27] Craig E. Wills, Gregory J. Snyder, and Christopher Kmiec. Persistent information retrieval on the Internet. In *Proceedings of the IASTED/ISMM International Conference on Intelligent Information Management Systems*, pages 152–155, Washington, D.C., June 1995.

[28] Lynn Ziegler, Adrienne Bloss, Dorothy Deremer, and Craig E. Wills. Cooperative learning in computer science. In *ACM SIGCSE Conference*, Nashville, TN, March 1995. Accepted workshop.