TEACHING COMPUTER SCIENTISTS TO PLAY WELL WITH OTHERS

Panel

Charles Welty [1]
Professor of Computer Science
Computer Science Department
University of Southern Maine
Portland, ME 04104
(207) 780-4240
welty@usm.maine.edu

Jesse M. Heines
Professor & Undergraduate Coordinator
Dept. of Computer Science
University of Massachusetts Lowell
Lowell, MA 01854
heines@cs.uml.edu

Margaret Menzin
Professor of Computer Science & Mathematics
Simmons College
300 The Fenway
Boston, MA 02115
menzin@simmons.edu

ABSTRACT

Most undergraduate computer science programs include classes that require team work. This helps our students work well with each other, but does not address the problem of working well with people from other disciplines. Computer scientists have preconceived notions of people in other professions and people in other professions have preconceived notions of computer scientists. These preconceptions can interfere with good working relationships. Computer scientists tend to work on projects of use in an application field that may be unknown to them and, thus, must work with people in that application field.

Many computer scientists enjoy the comic strip Dilbert™ by Scott Adams. A problem with the strip is that it paints stereotypical portraits of the various professionals that Dilbert works with. The marketing people lie and make impossible demands. Management is composed of idiotic, power mad people. Advertising people will promise anything. The human resources department is out for your blood. Certainly graphic artists and usability professionals would come under similar fire if they are ever part of the strip. Of course, the computer scientists (or engineers) are also negatively stereotyped as having no lives, being obsessed with hi-tech toys, having poor social skills

[1] Contact person.

and, generally, being geeky (see http://en.wikipedia.org/wiki/Geek). The problem is that these stereotypes often contain a component of truth. Computer scientists and others must learn to look beyond the stereotype and see what a person can actually do.

This panel will look at several ways to foster appreciation of other disciplines to help broaden the sometimes narrow perspective of our graduates.

PANELISTS' POSITION STATEMENTS

Charles Welty

Using a Graphical User Interface Design course to teach computer scientists to play well with others.

A course in graphical user interface design works well as a delivery mechanism to aid computer scientists to learn better what other people do in designing and building a product. Many assignments and projects in computer science courses start with the functionality and appearance of the software defined in the problem statement. On the other hand, part of our GUI course is deciding what product to produce and then going through cycles of design and user testing to determine the final form of the product. In this process students should learn what team members with backgrounds other than computer science bring to the table.

Our GUI course includes presenters with backgrounds in marketing, graphic arts, usability, ethnography, technical writing and, occasionally, management. Presenters from these other disciplines come into class not to teach the computer science students to be able to do the presenter's job after a single presentation but to show the breadth and depth of the presenters' expertise. Then, when the two come in contact at the workplace, the computer scientist has more than the above mentioned stereotypical views.

Jesse M. Heines

Exploring interdisciplinary course models.

Like our colleagues at the University of Southern Maine, we at UMass Lowell believe that there is great value in exposing computer science students to the work of their peers in other disciplines. With the help of a National Science Foundation (NSF) grant, we have tried to go beyond bringing in presenters from other disciplines to give students concrete experience in working on interdisciplinary projects.

In 2007, a team of UMass Lowell professors was awarded an NSF CPATH grant to explore the intersection of computer science and the arts through interdisciplinary courses. The team consisted of two professors in computer science, two in art, one in music, and one in theatre. This team developed two types of courses: "synchronized" and "hybrid."

"Synchronized" courses pair two existing of upper-level courses for majors in two departments. The courses remain independent, but the students work together on a joint project developed within the scope of the two courses. "Hybrid" courses are ones that are taught by two instructors simultaneously, one instructor from computer science and the other from the arts. These courses are open to all students across the university and co-listed in two departments. Science students earn Arts & Humanities General Education ("GenEd") credit, while Arts students earn Technology GenEd credit.

Not unexpectedly, each course model turned out to have pros and cons. Our work taught us a number of lessons that we are now using to revise the courses. As we

move through our third year of this project and approach the end of our NSF funding, we feel that we have laid sufficient groundwork for at least the hybrid courses to continue to be offered. Enrollments in these courses has increased over time, and the professors involved remain enthusiastic about working together. The university administration has embraced the effort and allowed the professors to count the joint teaching as part of the professors' normal teaching load.

We see these developments as indicators of a successful program that we expect to continue to grow and help prepare students for the interdisciplinary project teams they will encounter after graduation in the workplace.

Margaret Menzin

Working with a course in another discipline

In my Systems Analysis course I try to present students with opportunities to both interview high level stakeholders and to work with domain experts from another field. Last year one project in the course was the design of an informational website for prospective students. My students had to identify the stakeholders and then write to and interview them: many high level administrators at our university (deans, head of the Library, etc.) and groups of other students to determine what content and functionality should be on the site. We also invited to our class people such as the Vice President of Marketing, who spoke about the institution's approach to presenting ourselves and answered student questions.

After students had determined the architecture of the site we turned to the Video Production course in our Communications Department to provide content on a specified list of topics. Videographers are creative people and they can produce "fun" content, wonderful videos with humor and "voice". Creative people, however, don't necessarily follow your specifications. After the students in the Video Production course had made more than a dozen wonderful videos, my students had to turn to other sources to find images and videos for the remaining topics which needed visual material. There were also negotiations with the videographers about whether or not certain scenes in some videos were appropriate for the new site. And there were some other tensions around maintaining a time schedule (which my students had developed on Open Project), a problem well known in all software development.

In retrospect my students and I didn't get enough "buy in" from the Communications students, and failed to communicate to them the nuances of the site's purpose, while the Video Production students didn't get enough of an understanding of what it means to be on "an assignment". I think some of these problems could have been solved by scheduling the courses at the same time so that the students could have worked more closely together. On the other hand, both groups of students gained an enormous appreciation of what the others had to contribute to a project and what it means to collaborate across professions. As a measure of the success, we are planning future collaborations between the two courses.

ABOUT THE PANELISTS

Charles Welty, professor of Computer Science at the University of Southern Maine for 30 years, is interested in graphical user interfaces and interface usability. An

NSF grant originally funded the lab used in the GUI Design course and other courses. He has been working in the field of usability and, earlier, human factors since 1977.

Jesse M. Heines has been on the UMass Lowell faculty for 25 years after 10 years at Digital Equipment Corporation. He has a keen interest in computer science education and computer applications in the arts, particularly those in music. This interest is currently supported by an NSF grant in which two CS professors are teaching interdisciplinary courses with professors in Art and Music. Jesse's teaching focuses on the implementation and evaluation of interactive, user-centered programs with rich graphical user interfaces (GUIs), particularly those employing Dynamic HTML, JavaServer Pages, and XML and XSL and their related technologies. Jesse has a long record of applying and evaluating these techniques in educational settings.

Dr. Margaret Menzin is Professor of Mathematics and Computer Science at Simmons College in Boston. Her interests in computer science focus on database systems, web services and web centric programming (for which she maintains an extensive on-line annotated bibliography), systems analysis and health informatics. She also has long standing interests in both pedagogy and encouraging women to pursue careers in mathematics and science.

PANEL TIMING

First, there will be a brief introduction to the panel and panelists (3 minutes). Each panelist will then present their position for at most 12 minutes, allowing about 5 minutes for questions following each presentation and 15 minutes for questions after all the presentations.