HOW DO FACULTY PARTNER WHILE TEACHING INTERDISCIPLINARY CS+X COURSES: MODELS AND EXPERIENCES *

Darakhshan J. Mir
Bucknell University
d.mir@bucknell.edu

Sumita Mishra
Rochester Institute of Technology
sumita.mishra@rit.edu

Paul Ruvolo
Olin College of Engineering
paul.ruvolo@olin.edu

Lori Pollock
University of Delaware
pollock@udel.edu

Sam Engen
Bucknell University
spe002@bucknell.edu

ABSTRACT

As Computer science (CS) plays an increasingly significant role in many other disciplines, it is crucial for us as CS educators to create authentic interdisciplinary learning experiences for students. To better inform the design of such learning experiences, we sought to catalogue how faculty from both CS and other disciplines are currently collaborating to create such experiences. Specifically, this paper describes knowledge-seeking activities carried out through designing and implementing a workshop program that brought together twenty-four faculty with experience in partnered teaching of CS+X courses. The goal is to take the initial steps towards preparing and supporting CS faculty to create interdisciplinary CS+X courses through partnerships with faculty in other disciplines, in order to spur interdisciplinary thinking in students.

INTRODUCTION

Computational approaches are becoming increasingly prevalent in a number of engineering, scientific, and humanities disciplines. For example, data science is used in social network analysis, digital humanities, relationship tracking, genomics, authorship attribution, and learning analysis. Simulation is used in optimization, economics, and the science fields for climate modeling and protein folding. Further, truly transformative work at the intersection of computing and these other disciplines requires teams of people with heterogeneous skill-sets (both computational and non-computational) who, despite their differences in training, can work collaboratively. As computer science educators, the growth in the importance of computation in other fields presents us with a crucial challenge: how can we best prepare our students (both majors and non-majors) to work effectively on interdisciplinary applications of computer science? While far from the only answer, some educators are addressing this challenge (at least in part) through developing courses that emphasize computer science content, content from another discipline, as well as content that integrates across the two disciplines.

This paper describes knowledge-seeking activities towards gathering information on existing models of partnered teaching of interdisciplinary computer science + discipline X courses, where X includes a broad set of disciplines, including and, beyond the sciences. We describe the steps taken to gather and synthesize knowledge, experiences, and unanswered questions and challenges about preparing and implementing a truly interdisciplinary CS+X course through partnered teaching.

BACKGROUND

To be successful in partnered teaching, faculty need to address several challenges not found in a typical classroom, e.g., how to establish a partnership with faculty from another discipline, how to facilitate learning of knowledge and skills across the disciplines given diverse student backgrounds, and how to best guide students through open-ended interdisciplinary projects in CS+X. Courses in the spirit of the vision of truly interdisciplinary courses in CS+X (both at the graduate and undergraduate levels) have been developed at several institutions [4,5,7,8,17]. For example, Dodds et al. [7] investigated integrating an introductory computer science course with an introductory biology course and envision an era of “integrative” CS+X science education. Yanco et al. [17] created the Artbotics program-a collaboration between artists and computer scientists that uses robotics to teach computer science to undergraduates, culminating in projects that were exhibited at a local museum. Typically, any faculty who wants to prepare to create such a course by leveraging others' experiences currently has to perform the arduous task of researching who has taught a similar course, examining their materials, talking with them, and then adapting these materials to their own situation. For example, Barr et al. [2] discuss how CS faculty at three liberal arts colleges reached out to other disciplines to enable students' contextualized learning of computing. In a panel discussion, Settle et al. [15] summarized the challenges and lessons learned from their attempts at integrating computing in both the secondary and undergraduate curriculum in disciplines beyond CS.

Our goal is to gather information on some fundamental pedagogical considerations that faculty navigate when involved in partnered teaching of CS+X courses. Toward this end, we are interested in two sets of research questions.

Research Questions

Our research questions focus on understanding the design space that CS+X courses currently span. Specifically, we sought to map this design space along two principal dimensions. The first dimension considers the student audience and experience in these courses.

- Which disciplines did the students come from?
- What did the students seek from the course?
- What are the salient differences in pedagogical approaches between teaching students in a CS+X courses and more traditional courses?
- How did the faculty team assess student learning?

A second set of questions centers on the faculty teaching team and their journey to developing, delivering, and sustaining the course.

- What motivated the teaching team to develop the course in the first place?
- How did the teaching team form?
- What was the prior level of preparation of participating faculty in terms of cross-disciplinary knowledge?
- Did faculty encounter any administrative roadblocks in developing the course?

METHODOLOGY

We developed and implemented a two-day faculty workshop in October 2015 in which 24 faculty with experience in partnered teaching in CS+X were brought together. Participant recruitment included postings, emails, and literature surveys to identify CS faculty and X faculty, who had partnered to create and implement CS+X courses. Along with computer science, other disciplines represented at the meeting included biology, music, law, mathematics, engineering, chemistry and communications [1, 6, 7,8, 9, 11, 14, 15, 16]. When possible, teaching partners from a CS+X course attended the workshop together. A comprehensive workshop plan was developed by the project team over the course of three to four months to create a format that gathered aggregate knowledge and experience, as well as provided useful sharing and inspiration to all attendees.

In order to answer our research questions, we asked each attendee to describe their course through two lenses: the student experience and the faculty experience. The aggregation of these descriptions into a map of the design space of these courses, described in this document, was created via post hoc analysis of these descriptions by the workshop organizers. We collected the student experience and faculty experience data through poster templates that attendees filled out at the workshop. The templates served to both organize the participants' responses along relevant dimensions, and provided an artifact that could be shared with other participants during the workshop. The first poster-template asked participants to describe the student experience in their CS+X

course. It included questions on (1) the disciplinary background of the students in their course, (2) student learning outcomes, (3) course design considerations, and (4) course timeline in terms of activities and student learning progress. In small groups, the participants worked to extract generalizable lessons from the student experience posters, followed by whole group discussions.

The same exercise was repeated to catalogue and share their faculty experience. Participants filled out poster templates outlining: (1) who taught the course, including briefly description of each faculty member's relevant background and how they shared the classroom teaching, (2) reasons for choosing to teach the course and the mechanisms through which they found a partner, (3) course planning, highlighting any special considerations due to being an interdisciplinary course, (4) administrative challenges, such as the place of the course in the curriculum, teaching credit for both faculty partners, and sustainability of the course, and (5) faculty experiences during the semester, pedagogical affordances and challenges as the course was being taught, and student assessment.

STUDENT EXPERIENCE

This section provides a synthesis of the faculty members' responses to prompts regarding the student experience in their own course. This synthesis provides a first cut answer to the first set of questions in the "Research Questions" subsection.

Enrolled Students

As per faculty responses, the majority of students enrolling in the CS + X courses were either CS majors or majors in the discipline X; for example, students taking a CS + Biology course were mostly either CS majors or Biology majors. Some signed up for the courses to acquire a required general science or arts credit without having to stray too far from their home discipline, but many enrolled in the courses due to interest in the interdisciplinary subject matter. Of the courses studied, only two included graduate students. One had CS students and X students meet separately, with a few joint sessions including both classes.

Learning Outcomes

Three broad categories of courses emerged from the responses we received. Undergraduate level CS courses taught with another STEM discipline (CS + STEM) were the most common, and usually involved the application of basic computer science concepts and simple programming to analyze and synthesize lab data pertaining to X. These courses had an emphasis on problem solving techniques, building a foundation of CS and X knowledge, and interdisciplinary communication. Undergraduate CS courses taught alongside a discipline in the arts (CS + Arts) were focused on the applications of computing in creating art pieces, learning to work and manage time as part of a team, and presentation and peer review skills. Graduate level CS + X (grad CS + X) courses examined the intersection and consequences of computing and another discipline through

writing and discussion, and had a strong emphasis on communicating at a high level using common language.

Course Design Considerations

Most participants that had taught CS + STEM courses indicated that scaffolding and structure was an important facet of their course design. Most also indicated that they strove for broad knowledge of the subject area, rather than in-depth study of either discipline. Several participants mentioned giving timely feedback early in the course, deeming it to be especially important for courses requiring two sets of foundational knowledge. A few faculty indicated that "metacognition" and "problem solving" were key focuses of course design.

The CS + Arts courses had a less structured grading format focusing on the completion of long-term projects, and used a studio art schedule of long class sessions emphasizing hands-on experience rather than a lecture-lab schedule.

Overall, most courses placed more weight on labs or projects than on declarative knowledge such as quizzes to grade work done between exams. Another goal was to improve communication between the two disciplines by providing opportunities for CS majors struggling with X to get help from an X major, and vice versa. To foster interdisciplinary teamwork many courses integrated the material from both disciplines towards a common theme as much as possible.

Course Timelines

Several of the CS + X teams indicated that some students, typically those with no CS or X background, were overwhelmed early on by the rapid introduction of basic CS and X concepts integral to the rest of the course. Near the middle of the course, the focus generally shifted towards application of concepts to projects or labs. Almost all courses mentioned a high degree of student fatigue and worry about performance near the end of the term, when larger projects and exams were introduced. However, most timelines also indicated that students were surprised, excited, and proud of their accomplishments when the courses concluded.

FACULTY EXPERIENCE

This section provides a synthesis of the faculty members' responses to prompts regarding the faculty experience in developing and delivering their own course. This synthesis provides a first cut answer to the second set of questions in the "Research Questions" subsection.

Teaching team composition

Most teams consisted of one CS faculty member and one X faculty member. Several of the teaching teams included more than two faculty members (the largest being a team of four). The two courses that did not fit this pattern had a single instructor, however, in

this instance guest speakers from other disciplines were brought in throughout the course. While the disciplinary makeups of the teaching teams were heterogeneous, in many cases the team was composed of members that had done work, in some context, at the border of their home discipline and the other discipline. In almost all cases the teaching team formed in a bottom-up fashion (e.g., through personal relationships), rather than a top-down fashion (e.g., through a departmental or institutional directive). Teaching team members typically had pre-existing relationships with each other that informed their collective decision to develop and teach the course. In part, this observation may explain why many of the CS+X courses in our sample were taught at small colleges. Smaller institutions typically have tight knit faculty, and this cohesion provides faculty with opportunities to build personal relationships with faculty from other departments.

While the motivations for teaching a CS+X course were unique to each participant, the participants' responses tended to cluster into distinct themes such as: desire for personal development and intellectual curiosity, student development, and extrinsic factors, such as the need to offer general education courses in computing for non-majors, supporting a departmental or institutional initiative or strategic plan, and increasing the number of majors or minors in their department.

Course Planning and Design

Developmental milestones for CS+X courses in our sample tended to include a number of additional milestones that are typically unnecessary for traditional course development. For example, due to the significant administrative roadblocks (see next subsection), some teaching teams began the process of course development by securing external funding. This funding was crucial in making the development of the CS+X course, and the significant resources that the development of the course entailed, more palatable to the administration. A second example was sitting in on classes taught by other teaching team members. This activity was used as a way for faculty members to build familiarity across disciplines and to establish a common vocabulary for subsequent course development.

For the CS+X courses in our sample, traditional course development milestones were subject to many special challenges. For instance, depending on the particular blend of CS+X, there were few if any existing curricula on which to model the course. This was particularly acute in cases where the course content was not about blending two already well-established subjects, but instead about developing a synthesis of a new, integrated discipline (e.g., a course on the “science of information”). A second challenge specific to developing a CS+X course was the need for faculty to learn new content, techniques, and language from the other disciplines in order to begin the process of course planning.

When designing class activities, a number of challenges arose that would be atypical in a traditional, disciplinary course. First, developing assignments and projects for students with a wide variety of backgrounds, skills, and modes of thinking presented a very difficult educational design problem. A commonly expressed hallmark of successful assignments in CS+X courses was allowing for every student to do work in both CS and X. This is a difficult challenge to meet when students bring to the course vastly different

levels of preparation in either CS or X. This goal is made more difficult for team-based assignments where students may gravitate towards doing work on the part of the assignment that they already know how to do well. Another challenge in designing classroom experiences was integrating activities that are alien to one discipline but essential to the other. For example, in a course that blended CS and studio art the idea of an in-class critique, while quite unusual in CS, is an essential component of studio art classes. This disciplinary friction was something that had to be hashed out both by the faculty during the development of the course and by the students during the semester. A final challenge that was cited more than once was the difficulty of framing a class in such a way that it was compelling to multiple student audiences. Towards this goal, one participant expressed that one has to get past the idea that interdisciplinary material is compelling simply by virtue of its interdisciplinary nature, instead one should create educational activities that provide students from multiple disciplines with new skills that they can carry forward as they continue their studies within their major.

Administrative Challenges

The workshop participants commonly encountered administrative challenges in developing, delivering, and sustaining their CS+X course. Interestingly, there were no reports of administrators doubting the worthiness of the proposed courses. Instead, difficulties broke down roughly into two categories. The first category was related to the fact that the courses were co-taught. There were numerous reports of difficulties with having the course count fully toward each of the teaching team members' teaching loads. In a number of cases, due to external funding for course development, the course was able to count for both faculty members for a limited number of offerings, however, in these cases offering the course became difficult to sustain once the funding ran out. These challenges were exacerbated for departments that were experiencing significant over-enrollment (e.g., computer science departments in the mid 2010's). A second class of problems arose at the departmental level -- the issue of which requirements the course would fulfill was contentious for a number of the courses in our sample. In some cases, the faculty members had to promise to cover a specific list of topics in order to appease department heads, a constraint that could be difficult to fulfill while also developing the new course.

Experience During the Semester

Participants described a number of challenges that arose while delivering their courses. Firstly, there were challenges that arose simply due to the fact that they were teaching with another faculty member, e.g., divvying up class time, standardizing grading, and adjusting to stylistic differences in teaching. A related set of challenges developed on the student end - students often had trouble dissociating the content of an individual class from the faculty member that delivered it. Specifically, some students had difficulty viewing content as interdisciplinary, when disciplinary faculty members were the ones delivering it.

A number of faculty members cited challenges related to the different student majors that took their course. In most cases, while the courses were interdisciplinary, the

students were enrolled in traditional disciplinary majors. One faculty member cited that a crucial shift that happened during the semester was thinking of each class activity through the lens of each of the student majors. That is, each class activity should have a takeaway for students majoring in CS and students majoring in discipline X. A related issue was the need to foster a safe and supportive classroom environment that worked for all students. Specifically, the tendency for students being afraid to ask questions for fear of looking foolish may be magnified in a CS+X course. Additionally, for topics such as programming, it was easy for non-CS students, who may be learning to program for the first time, to conclude that they are intrinsically unsuited to the activity if they compared themselves to CS students.

LESSONS AND RECOMMENDATIONS

For those seeking to create a CS+X course, our work can be used in a number of different ways. First, the experiences described here (e.g., partnering with other faculty, designing a course, delivering a course, etc.) can be used to understand which activities have been successfully battle tested, which ran into problems, and what strategies were employed to mitigate encountered difficulties. We hope that through our synthesis of important considerations in teaching CS+X courses, that interested faculty will be able to begin the task of course design from a more informed place than would have been possible without our work.

Second, the poster templates used in the workshop can be used to help in the process of brainstorming and designing potential new courses. Three templates can be downloaded from our website: link-redacted-for-blind-review. The two templates on the faculty and student experience can be used to sketch out a new idea for a course, or to unpack the main ideas in the existing courses of two faculty members that want to teach together. A third template, which combines some elements of both the faculty and student experience templates, was explicitly designed for sketching out a new course.

CS+X courses present a rich set rewards and corresponding challenges for both faculty and students. As the world increasingly demands that computer scientists look outside of their field to engage with difficult interdisciplinary problems, the importance of these courses will only increase. We hope that our work serves as a useful resource as more and more faculty take up this important challenge.

ACKNOWLEDGMENTS

This material is based upon work partly supported by the National Science Foundation under Award DUE- 1456275. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors would also like to thank the attendees of the faculty workshop who provided the data for this work.

REFERENCES

[1] Barr, V. 2012. Create Two, Three, Many Courses: An Experiment in Contextualized Introductory Computer Science. *Journal of Computing Sciences in Colleges* 27.6 (June 2012), pp. 19-25.

[2] Barr, V., Liew, C.W., and Salter, R. 2010. Building Bridges to Other Departments: Three Strategies. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education (SIGCSE '10)*. ACM, 64-65.

[3] Bates, R., Goldsmith, J., Berne, R., Summet, V., and Veilleux, N. 2012. Science fiction in computer science education. In Proceedings of the 43rd ACM technical symposium on Computer Science Education (SIGCSE '12). ACM, 161-162.

[4] Carter, L. 2014. Interdisciplinary Computing Classes: Worth the Effort. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14)*. ACM, pp. 445-450.

[5] Cushing, J.B., Hastings, R., and Walter, B.L. 2009. CSO++ Broadening Computer Science at the Entry Level: Linguistics, Computer Science, and the Semantic Web. In *Journal of Computing Sciences in Colleges*. 25.1 (Oct. 2009), pp. 135-142.

[6] Danyluk, A., 2004. Using Robotics to Motivate Learning in an AI Course Aimed at Non-Majors. In *Proceedings of Spring Symposium on Accessible Hands-on Artificial Intelligence and Robotics Education* (AAAI 2004).

[7] Dodds, Z., Libeskind-Hadas, R., and Bush, E. 2012. Bio1 As CS1: Evaluating a Crossdisciplinary CS Context. In *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '12)*. ACM, 268-272.

[8] Greenberg, I., Xu, D., and Kumar, D.. 2012. Processing: Creative Coding and Generative Art in Processing 2. Apress.

[9] Downey, Allen B., 2014. Think Stats: Exploratory Data Analysis in Python, V 2.0.30, Green Tea Press.

[10] Gentile, L., Caudill, L., Fetea, M., Hill, A., Hoke, K., Lawson, B., Lipan, O., Kerckhove, M., Parish, C., Stenger, K., Szajda, D., 2012. Challenging disciplinary boundaries in the first-year: A new introductory integrated science course for STEM majors. *Journal of College Science Teaching*. Volume 41, Number 5..

[11] Greher, G.R. & Heines, J.M. 2014. Computational Thinking in Sound: Teaching the Art and Science of Music and Technology, Oxford University Press.

[12] Guzdial, M., and Soloway. E., Teaching the Nintendo Generation to Program. 2002. In *Commun. ACM* 45.4, 17-21.

[13] Kurose, J., Ryder, B., Kelemen, C., et al. 2005. Report of NSF Workshop on Integrative Computing Education and Research. In Northeast Workshop ICER.

[14] Metaxas, P.T., and Ribner N. 1999. Experiences from teaching an interdisciplinary multimedia course. *IEEE International Conference on Multimedia Computing and Systems (ICMCS'99)*. IEEE, 586-591.

[15] Settle, A., Goldberg, D.S., and Barr, V. 2013. Beyond Computer Science: Computational Thinking Across Disciplines. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*. (ITiCSE'13).ACM, 311-312.

[16] Tartaro, A., Chosed, RJ. Computer scientists at the biology lab bench. 2015. In *Proceedings of the 46th ACM Technical Symposium On Computer Science Education (SIGCSE '15)*. ACM, 120-125.

[17] Yanco, H.A., Kim, H.J., Martin, F.G., and Silka, L., Artbotics: Combining Art and Robotics to Broaden Participation in Computing. In *AAAI Spring Symposium: Semantic Scientific Knowledge Integration*. 2007, p. 192.