Team-Teaching with Colleagues in the Arts and Humanities

Panel

Keith J. O'Hara
Bard College
Annandale-on-Hudson, NY, USA
kohara@bard.edu

Sven Anderson
Bard College
Annandale-on-Hudson, NY, USA
sanderso@bard.edu

David Musicant
Carleton College
Northfield, MN, USA
dmusicant@carleton.edu

Amber Stubbs
Simmons College
Boston, MA, USA
stubbs@simmons.edu

Thomas Way
Villanova University
Villanova, PA, USA
thomas.way@villanova.edu

KEYWORDS

Team-Teaching; Interdisciplinary Courses; Computing for All; Humanities Computing; Arts

1 SUMMARY

This panel will include experience reports from five computer science faculty members who have team-taught courses with professors from outside the sciences. Specifically, we will discuss lessons learned and best practices with collaborating with faculty from the arts and humanities. Courses that look outward have the potential to broaden participation and promote computing's role in the broader world beyond software engineering concerns. The panelists will highlight how to: find a topic, find a collaborator(s), design the course, maintain rigor in both disciplines, target the right audience, assess how well it worked, and do it more than once.

2 SVEN ANDERSON

I collaborated with a professor of literature, Collin Jennings, to develop and co-teach a course entitled "Technologies of Reading: Human and Machine Approached to Literature." This course assumed no background in computing and was attended primarily by students in literature. We sought to highlight the history of scholarly activity that included both humanist and computational approaches to literary analyses, particularly ideas associated with close reading. The course was organized historically, and used computation to re-create and then extend analyses performed in the papers on literary analyses we read, combining basic elements of computation with more powerful algorithms adopted from the

Natural Language Toolkit, including parsing, sentence generation, language modeling, and creation of high-dimensional semantic spaces. I will focus on how we determined the arc of the course topics, how we balanced the pull of literary studies versus basic programming. The long-term prospects of the course and its effect on later teaching by both professors will be discussed.

3 DAVID MUSICANT

Stephen Mohring (Professor of Art, Carleton College), and I have offered a course three times called "Art, Interactivity, and Robotics." In this hands-on studio centered course, we explore and create interactive three-dimensional art. Using basic construction techniques, microprocessors, and programming, this class brings together the fundamentals of computer science, sculpture, engineering, and aesthetic design. Students engage the nuts-and-bolts of fabrication, learn to program computers, and study how robots think. The class culminates in a campus-wide exhibition. I will describe my experience from having taught the course three times, specifically regarding lessons learned in marketing an interdisciplinary course such as this one, as well as how we faced challenges in producing assignments in a course that introduces dramatically different sorts of skills from multiple disciplines.

4 AMBER STUBBS

Kris Erickson (Communications Department, Simmons College) and I will offer a "learning community" course titled "Choose Your Own Adventure: Coding and Digital Storytelling." Learning Communities are part of Simmons College's effort to incorporate multidisciplinary learning into every student's curriculum by bringing together different fields of study. In our Learning Community, students will learn programming fundamentals (variables, if/else, loops, arrays, functions, etc, using JavaScript) and simple HTML and CSS, along with basic photography, photo manipulation, and video processing to create immersive, interactive, and responsive stories through. Finding a balance between integrating and maintaining rigor in both disciplines can be difficult, but there are ways to use each discipline to motivate the other. I will discuss how Professor Erickson and I developed our current version of the course, which uses individual and team-taught classes to provide instruction, as well as related work using the Twine platform (twinery.org) to teach programming with the Boston chapter of Science Club for Girls.

5 KEITH O'HARA

With Ben Coonley (Film & Electronic Arts Department, Bard College), I offered an 8-credit course called "Games at Work: Participation, Procedure and Play." The course was part of a campus-wide initiative to create "big ideas" courses that connects programs in different divisions of the college. These courses targeted first and second-year students, and therefore, did not require any prior coursework in the involved disciplines. This single course satisfied two college-wide general education distribution requirements: Mathematics & Computing and Practicing Arts. Students created a variety of digital and non-digital games: remixing classics like Candyland and Fluxus Event Scores; programming with Twine, P5/JavaScript and Unity. I will describe my experience with an intimate (15 students; 2 faculty members), intensive (6 hours/week) course including challenges (e.g., time management, scalability, sustainability) and successes (e.g, student engagement, improved pedagogical practice, incubating continued collaboration). Moreover, I will compare this model of intensive team-teaching with other interdisciplinary models like linked/conjoined classes[1] and module-based approaches, both of which I have employed at Bard with humanities and art collaborators.

6 TOM WAY

Two French professors in my university's Department of Romance Languages & Literature, Seth Whidden and later Francois Massonnat, and I collaborated for the past six years on a parallel, conjoined approach to team-teaching[2].

We taught courses for our majors and minors on the same days and times in adjoining classrooms, meeting together about 10 times a semester to explore shared topics of interest and work together on projects. French students were in their Writing and Stylistics in French course while Computer Science students studied in my Machine Translation course. Motivated by a university initiative to foster interdisciplinary teaching, we designed our approach because we wanted to find a fun and fulfilling collaboration while maintaining rigor within our respective disciplines. I will report on how we pleased our chairs and dean by overcoming the "who gets credit for how much teaching" hurdle and on the successful student and faculty collaboration that resulted. Presented will be some of the interesting language analysis projects devised by cross-disciplinary teams of students using NLTK, along with the many positives and few negatives we gleaned through three iterations and our plans for the next offering in Fall 2018.

REFERENCES

[1] Keith J. O'Hara. 2012. Pedagogical Explorations in Computational Perception for Performance. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI'12)*. AAAI Press, 2375–2376. http://dl.acm.org/citation.cfm?id=2900929.2901069

[2] Thomas Way and Seth Whidden. 2016. A Parallel, Conjoined Approach to Interdisciplinary Computer Science Education. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '16)*. ACM, New York, NY, USA, 363–363. https://doi.org/10.1145/2899415.2925486