Evaluation of the Conjoined Approach to Interdisciplinary Computer Science Education

Thomas Way
Department of Computing Sciences, Villanova University, Villanova, PA, USA

Abstract - *As interdisciplinary education has gained in popularity, educators are finding advocates in academia and industry. There is widespread recognition that combining content from computer science with disparate disciplines enables students to gain exposure to real world collaboration in an appealing and valuable way. Implementing a truly interdisciplinary course, where two or more disciplines are merged into a single course, is energizing but also comes with many challenges. In this paper, we present an effective approach that attempts to minimize the difficulties often encountered in creating an interdisciplinary course. We describe our parallel, conjoined approach in the context of interdisciplinary teaching, review its logistics, analyze quantitative and qualitative longitudinal data covering three successive offerings, and introduce a number of examples of specific successful project collaborations.*

Keywords: Interdisciplinary courses, parallel conjoined approach, distributed expertise, machine translation, natural language processing, writing, stylistics.

1 Introduction

Computing has increasingly woven its way into the fabric of daily life. Omnipresent adoption of computer technology into virtually every human endeavor has emphasized the need for educating a generation of computer literate citizens in a new age of interdisciplinarity. Specialization is no longer sufficient as nearly all fields of human activity require an understanding and application of that field within the context of computing and often additional fields. Within education, this interdisciplinary vision is accomplished by combining of two or more distinct and contrasting disciplines into a single, cross-discipline learning experience. [8]

The need for strong interdisciplinary understanding extends beyond academia to industry, and to society in general, and involves crossing traditional topic boundaries to explore novel combinations of historically disjoint fields. [8]. These new topic groupings may be inherent, as when a new program of study or research arises, such as nanotechnology which combines ideas from physics, chemistry, and mechanical and electrical engineering [11]. They also be less obvious, as with research that uses machine learning to classify the social behavior of dairy cows using accelerometer-based technology [7].

Breaking through the traditionally siloed culture of academic specialization can lead to a variety of increasingly well-recognized benefits. Interdisciplinary endeavors are uncovering synergies between approaches to teaching and learning among disciplines, settling differences of opinion regarding the importance of interdisciplinarity, and improving upon a host of institutional, administrative and logistic issues. [1, 4, 8]

At the university level, interdisciplinary education has traditionally led to the production of a single college course that tightly integrates two or more subjects, resulting in a new combined topic area that may not have existed before. This new course often represents considerable effort by two or more faculty who collaborate closely, contribute discipline-specific material, design cross-discipline content and exercises, and team-teach the course. The endeavor is motivated by optimistic anticipation on the part of faculty and administrators, soon tempered by the reality of just how difficulty creating and offering such a course can be. Faculty discover the process to be labor intensive, time consuming, and logistically cumbersome. Administrators can find that the initially appealing and promotable idea becomes difficult to harmonize within a generally inflexible system of rigidly defined course categories, tight budgets covering faculty time and materials, and institutional barriers to equitably distributing teaching credit among multiple faculty for the same course.

To contrast with our approach, we have dubbed the conventional, merged-topic, tightly-integrated approach described above as the *joined interdisciplinary* model. With a joined approach, the topics in an interdisciplinary course are integrated and overlapping, the result of careful design and significant effort, producing what amounts to a new discipline that merges elements of all contributing subject matter.

In this paper, we present the latest version of our *conjoined interdisciplinary* model. This approach attempts to overcome many of the organizational and administrative challenges inherent with the traditional, joined approach while still providing students with much of the same cross-disciplinary benefits of an interdisciplinary offering. We first describe the conjoined approach in the context of interdisciplinary teaching and review the logistics of the approach. Then, we analyze results of three iterations of applying the conjoined approach, and describe a number of examples of specific, successful project collaborations.

2 Background

Interdisciplinary computer science education is an active and ongoing area of pedagogical research with increasing relevance to faculty and students [1, 4, 8, 10, 12, 13]. The majority of work in this area has been using variations of the joined interdisciplinary model, as most work aims at designing a single course. This is not unreasonable, as it is a natural way to think about the design of a course that merges two or more disciplines.

2.1 Joined Model

Within an interdisciplinary course, there are at least two topic areas that are combined to produce the material covered by the course. In figures below, these two topic areas are labeled "A" and "B". For the joined model, and later the conjoined model, these labels can indicate two distinct topics, two faculty members, two pre-existing courses, or even two or more student populations.

The joined interdisciplinary model (Figure 1) unites topics A and B into a new topic C. This is the traditional and general way to think about and design an interdisciplinary course, whether or not in combination with computer science. Significant faulty effort goes into crafting a single course offering (indicated by the long arrow line) that thoroughly covers the amalgamated topic C. This diagram illustrates the combination of two topics into a new topic, the joining of efforts by two faculty members to prepare and present the new topic, and the combination of students with potentially many academic backgrounds into a single group all studying in course C.

Figure 1. Joined interdisciplinary model.

2.2 Distributed Expertise

The general classification of collaborative education, which includes interdisciplinary approaches, is *distributed expertise* (Figure 2). The distributed expertise model is a way to express the continuum of various forms of interdisciplinary collaboration between two cooperating experts. When collaborating, faculty members contribute to a lesser or greater extent, as dictated by the needs of the collaboration, from their own backgrounds. This model can be helpful when beginning an interdisciplinary course collaboration as a way to identify how two (or more) faculty shall contribute their expertise. [3, 13]

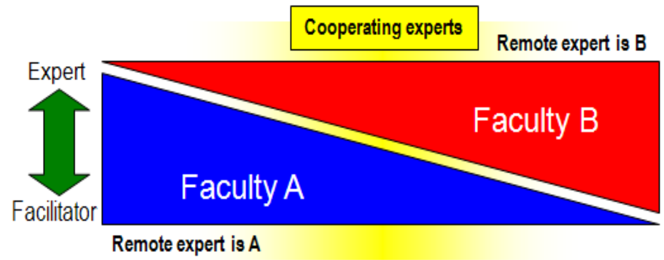

Figure 2. Digital distributed classroom expert-facilitator model continuum.

2.3 Collaboration

Team-teaching is an obvious approach to use when developing a course that draws material from disparate disciplines. Using a team-teaching model where an educator from computer science collaborates with an educator from another discipline has been successfully applied to a Project Based Learning (PBL) approach at the high school level [5].

There is evidence that having students collaborate within interdisciplinary project groups is very beneficial to learning at the university level and provides students with significant subject matter relevance [6]. While these approaches can be quite effective, they do not themselves overcome the administrative challenges of accounting for faculty effort and assigning course hour credit equitably in team taught courses at the university level.

2.4 Next Generation Interdisciplinarity

The literature provides a vast breadth and depth of experiential data from the use of variations on the joined interdisciplinary model that extends far behind computer science oriented collaboration. That interdisciplinary efforts continue and have been widely adopted is encouraging. A theme that is identified in much of the past work in this area, and one that motivates the innovations reported here, is that of the many challenges inherent in the joined approach. Acknowledging these challenges, and by mining the wealth of interdisciplinary course design experience of our colleagues, has led to the design and evaluation of our conjoined model, our novel approach to addressing the numerous hurdles to successful, interdisciplinary course design and education.

3 Conjoined Model

Clearly, there have been, and continue to be, many successful interdisciplinary courses developed and offered at colleges and universities, there also continue to be significant, unavoidable difficulties involved. While the work reported in this paper focuses on interdisciplinary collaboration between computer science and non-computer science disciplines, the model presented is equally applicable to the pairing of any disciplines.

3.1 Motivation

There is increasing and demonstrated value in computer science to other disciplines [3, 6], including the arts [9, 10], and a commensurate need for the development of computational thinking skills [16]. Computer science educators are uniquely positioned to fill this need. Thus, the goal of this research is to develop an interdisciplinary instructional approach that brings together students and faculty in two, traditionally non-overlapping, academic disciplines, one of which is computer science, and to do so in a way that minimizes the challenges and maximizes the chances for successful learning. Building upon that fundamental need is the motivation to reduce the time and effort needed to coordinate interdisciplinary offerings while maintaining their value to faculty, students and administrators.

Experience with previous efforts to create and offer interdisciplinary course models [1, 12, 13] led to an understanding of many of the difficulties inherent in the traditional, joined approaches. Thus, identifying the challenges and ways to overcome them motivated the design of a loosely-coupled interdisciplinary approach to course design. Applying concepts of distributed expertise and formulating practices that address the various challenges of a joined approach led to the design of the conjoined model.

3.2 Design

The conjoined model (Figure 3) differs from the joined model (Figure 1) in that it does not require the creation of a new topic C. Rather, the conjoined approach remains at its core two distinct topics, A and B. The innovation is that multiple topical "merge points" are incorporated during a course offering. The diagram illustrates how two previously existing courses, A and B, taught by two faculty members serving two distinct populations of students, can produce the effect of an interdisciplinary learning experience is a more efficient and less complicated way.

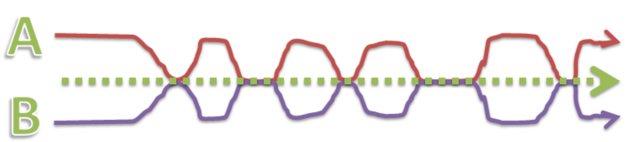

Figure 3. Conjoined interdisciplinary model.

In the basic use of this novel approach, there are two separate classes taught by two different faculty members and attended by two disparate student groups. The two courses run in parallel, meeting in classrooms near each other on the same days and times. This nearness facilities the coming together numerous times during a session (i.e., semester, quarter, etc.) to collaboratively study and learn mutual or overlapping topics of interest.

3.3 Logistics

To apply this model, there are six, well-defined steps to follow. First, it is necessary to **identify a collaborator**. Two interested faculty members recognize that there is interesting in collaborating. Second, the two **determine courses** that are appropriate for this collaboration, hopefully making use of existing ones they each already offer. Third, it is necessary to **acquire approval** to collaborate, which is primarily getting buy-in from each faculty member's department and dean, if necessary, and the appropriate administrative support including having the two courses scheduled to meet nearby each other on the same days and times. It is important that at least one of the two classrooms used has capacity enough for combined class meetings. Fourth, time is spent **planning the collaboration**, with the identification of specific points throughout a session when the courses would benefit from conjoined activities. Fifth, individual faculty spend time in **preparation** for offering their courses as they normally would, customizing selective content to prepare for the joined experiences. Finally, sixth, the session commences and the courses, as designed, are **delivered** to students.

3.4 Benefits

There are numerous benefits to the logistics of the conjoined model as compared with the joined model (Table 1). The most notable benefits to using the conjoined model are the dramatic reduction in risk and the easier accounting of faculty credit inherent in the approach.

Table 1. Comparison of joined and conjoined models.

Steps	Joined	Conjoined
Identify collaborator	complicated	less complicated
Determine courses	challenging	easy
Acquire approval	many details	easy
Plan collaboration	hard work	hard work
Preparation	lots of time	less time
Delivery	tricky	less tricky

There is always a risk that students or faculty may not realize the intended benefits of an interdisciplinary course, and with the conjoined model that risk is minimized. If it was determined that the collaboration is not working for any reason, the two courses could continue to run independently with the only impact being the lack of cross-discipline collaboration. Majors and minors still fulfill their requirements, and aside from disappointment that the interdisciplinary aspects were not explored, everybody is okay.

From an administrative perspective, since two existing courses are being taught by individual faculty members, no special accounting is needed to assign teaching credit. Aside from the requirement to schedule course to meet on the same days and times, hopefully in nearby classrooms (though this is not an absolute necessity), no other difficulties arise for administrators. The benefit to administrators is the lack of teaching credit headache coupled with being able to rightly claim that their departments are offering interdisciplinary learning.

The lighter weight approach to interdisciplinary course design means that identifying collaborators, selecting appropriate courses, acquiring approval for the effort, and pre-session preparation are all easier than with a joined interdisciplinary model. Of course, hard work is required to plan the collaboration, though the coordination of topic cross-over should be eased by the reducing coupling and integration of topics as compared with the joined model. Once a session beings, the courses tends to be less tricky to schedule and deliver since each instructor is already dedicated to teaching on those days and times.

4 Evaluation

The conjoined model was evaluated by combining two discipline-specific courses from very divergent subjects and repeating the collaboration three times over a six year period (Fall 2012, 2014, and 2016). Courses were held at the same class meeting times in adjacent classrooms and with regularly scheduled, combined meetings. These courses were “Machine Translation” offered as an elective course to Computer Science (CS) majors and minors, and “Writing and Stylistics in French” that is a required course for French and Francophone Studies (FFS) majors and an elective for minors.

The only administrative issues we dealt with were getting approvals from department chairs in our respective disciplines to combine occasional class meetings and requesting that the registrar schedule our meetings on the same days and times in nearby classrooms. Informally, we also received enthusiastic support on this approach from our students, colleagues, dean, and university president.

Our goals were to support departmental and university interdisciplinary initiatives, create a rich, educational experience for our students and ourselves, extend the FFS curriculum beyond its traditional scope, and develop a portable approach for combining CS topics with disciplines that traditionally might not be considered as compatible with CS. As anticipated, and previously reported [14, 15], the conjoined model proved to be a significantly less difficult path to follow in the design and implementation of an interdisciplinary course offering.

Course implementation involved between 8 and 12 combined course activities, with some pre-collaboration instruction given in each separate course to prepare students for the collaboration. Combined activities began with joint lectures that introduced students to overlapping concepts and to specialized topics from each discipline that had relevance to the collaboration. For example, concepts from writing and stylistics and from software design were presented, affording students from the CS and FFS, respectively, to gain introductory grounding in unfamiliar topics. As the semester progressed, interdisciplinary teams of student were formed to collaborate on a number of design projects where the expertise of students in one area would benefit students in the other area. A culminating activity resulted in cross-disciplinary projects that were developed by these teams with learning objectives that related to the specific needs of the two groups of students.

The effectiveness of the conjoined approach to our two course collaboration was measured using student performance on graded assessments, an evaluation instrument that students completed at the beginning and again near the end of the semester, and observations made throughout the semester. These measure provide a wealth of quantitative and qualitative support for the approach, as well as valuable feedback that has led to continued refinements to the approach. Data was collated and summarized for a total of 51 CS and 45 FFS students.

4.1 Student Performance

For the CS course, the average semester grade for the 51 students enrolled in the three iterations of the course was A-, with a grade distribution of: A (26), A- (20), B (3), B- (2). All assignments were completed by all students, and the variability in grading was due, not surprisingly, to thoroughness and consistency of the work done. The CS course was project-oriented, so the culminated exercise was a final project rather than an exam. Students participated enthusiastically and appeared to particularly enjoy the cross-discipline collaboration, with a common observation that they had never recognized how computer science could apply to analysis of something as “outside of computer science” as the French language.

For the FFS course, the average semester grade for the 45 students over three iterations was B+, with a distribution of: A (5), A- (10), B+ (15), B (13), and B- (2). All students completed all of their assignments, and the variability in grading was due to the differing degrees of preparation for class meetings, for the quantity and quality of participation in French when the FFS class met apart from CS, and the results of project collaboration. In two of the collaborations, a final paper reported on the culminating project, with students demonstrating a high degree of creativity and enthusiasm; the students appreciated the ability to pursue individual projects that were an extension of their existing interests. FFS students developed a deeper appreciation for computers’ abilities to analyze different parts of language, and as such they found a greater respect for computer science’s problem-solving function.

4.2 Evaluation Instrument

The evaluation instrument measured quantitative and qualitative perceptions of the students regarding various aspects of the combined course material. The four principal, quantitative measures were: skill level with computers in general, computer programming specifically, familiarity with the topic of "Writing and Stylistics in French" and the ways in which computer technology, in particular Machine Translation, can be used in conjunction with the study of Writing and Stylistics in French. Students self-evaluated their ability and familiarity for each of the measures on a 10 point scale, with 0 being lowest and 9 being highest. Table 2 shows the average values for the four categories measured.

Table 2. Overall average values per class for evaluation instrument measured at start and end of Fall 2012, 2014, and 2016 semesters.

	CSC		**FFS**	
Categories (0 is lowest, 9 is highest)	**start**	**end**	**start**	**end**
1. Skill level with computers	6.8	7.8	4.6	5.4
2. Computer programming ability	6.2	6.8	1.1	1.3
3. Familiarity with Writing and Stylistics in French	0.7	2.7	4.7	6.9
4. Uses of CS in Writing and Stylistics in French	1.6	5.7	2.5	6.6

Figure 4 provides a graphical comparison of the same data, illustrating student perceptions of what they knew at the start and end of the semester.

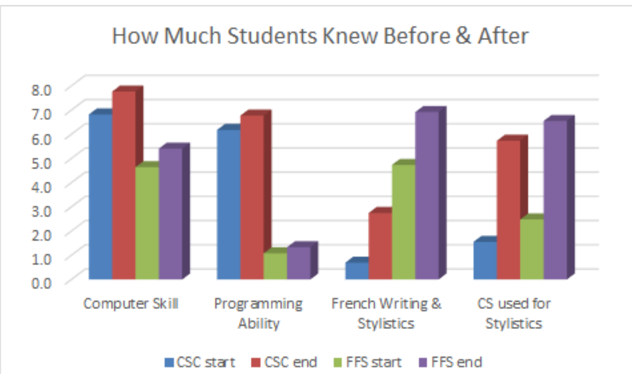

Figure 4. Comparison of student-perceived knowledge.

Improvement was observed in all areas measured (Table 3 and Figure 5), though some were slight. Both CS and FFS students showed similar and significant improvements in their understanding of stylistics (2.1 and 2.2) and the use of CS to analyze stylistics (4.2 and 4.1), while both groups underwent slight but measurable improvement in computer skill (0.9 and 0.8) and programming ability (0.6 and 0.2). These results differ slightly from previous evaluations [14, 15] where CS students over-estimated their computing abilities while FFS students under-estimate theirs, and is likely the result of more recent students having more exposure to computing throughout their education and so having a more accurate view of their current level of knowledge rather than a lack of learning.

Table 3. Improvement in evaluation instrument measures per class and combined from start and end of Fall 2012, 2014, and 2016 semesters.

	Improvement		
Categories (0 is lowest, 9 is highest)	**CSC**	**FFS**	**Overall**
1. Skill level with computers	+0.9	+0.8	+0.9
2. Computer programming ability	+0.6	+0.2	+0.4
3. Familiarity with Writing and Stylistics in French	+2.1	+2.2	+2.1
4. Uses of CS in Writing and Stylistics in French	+4.2	+4.1	+4.1

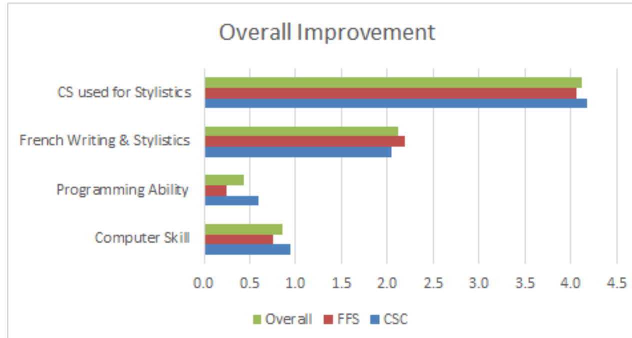

Figure 5. Comparison of student-perceived knowledge.

Among the more common qualitative written responses were CS students who remarked on noticeable and somewhat surprising appreciation and understanding of how CS can be applied to the analysis of writing. They demonstrated noticeable improvement in understanding of the subject matter of and uses of CS ideas for analysis of written language. These students observed that the experience demonstrated the value of "understanding the constraints of computing when applied to natural language" and "learning something new from an expert in a different area and applying that to my own software". The CS students found particular satisfaction in developing complex software tools that used their programming skills and creativity to assist students in a French course.

In the first two iterations of the collaboration [14], FFS students demonstrated a much clearer understanding of how CS can be used to analyze and more deeply appreciate French writing and style. In the third iteration, while FFS students expressed similar observations, a few expressed frustration with not feeling full part of the collaboration. This we believe was due to an overly strong sense of ownership of some CS students in the work being done and a corresponding reluctance to fully involve FFS students in hands-on programming activities. It was also apparent that the learning style of some of the FFS students in the third iteration was less conducive to the collaboration. In future offerings, these objections will be addressed by preparing both CS and FFS students more for the collaboration in anticipation of these difference in learning style and ownership.

4.3 Analysis

The results of our evaluation instrument and qualitative observations convinces us of the worth of the conjoined interdisciplinary approach. After three arguably successful iterations employing the conjoined model, we continue to feel that it truly does present a path to interdisciplinary computer science education that is less complicated and easier to implement than the more traditional joined interdisciplinary approach.

We benefited from offering the same course collaboration three times. One challenge was the switch to a new FFS instructor for the third iteration, which included a slight switch in teaching style and course details for FFS students. These differences proved to be minor, with a few alterations in the CS course needed to accommodate the differences in approach. In all cases, we observed many benefits to the faculty pedagogical process and to student learning. As faculty, we were able to move outside of our respective specialties to learn about and collaborate on teaching about material from the other's discipline. We experienced along with our students the excitement of collaboration and mutual discovery, as we observed CS students feeling empowered to offer suggestions and develop solutions using their disciplinary backgrounds and FFS students uncovering their computational thinking skills as they gained deeper understanding of how computer tools could be developed to solve the language analysis problems they wanted to solve.

The conjoined model does appear to reduce the challenges typically seen with joined interdisciplinary approaches making it particularly suitable to junior faculty who may be eager to apply the latest ideas in interdisciplinarity yet are wary of straying from their specialization. For any faculty members, using the conjoined approach can provide the benefit of simultaneously fulfilling departmental, college and university interdisciplinary initiatives or missions in a more manageable way.

5 Example Projects

To illustrate the concrete outcomes of using the conjoined approach, the following are collaborative projects that CS and FFS students produced.

Pastiche poetry - FFS students were tasked with gathering a number of samples of French poetry written by a selected, prominent French poet. Based on those samples, FFS students then wrote their own "pastiche" poem in the style of that famous poet. Meanwhile, CS students developed natural language processing and machine translation tools that could analyze French writing for various characteristics such as syllable count, word choice, and rhyming scheme, that can be indicative of individual, poetic style. A collaborative student was then done in class to analyze FFS student pastiche work and compare them programmatically with the samples upon which they were based to measure "nearness" and to quantify similarities and differences in style. FFS student work was of very high quality as verified both by the CS tools that found results to be very similar and by the FFS faculty member who manually compared the results.

Language translator evaluation - CS students were tasked with devising their own language translation tools to translate between English and French. Tools were a combination of statistics-based, sample-based and word-based translators that, while necessarily rudimentary due to constraints of time and student level, produced satisfactory, if imperfect, results. FFS students then evaluated the translation results, applying their familiarity with the French language in general and with writing analysis specifically. All agreed with the results of FFS student analysis that the translations were not nearly as good as what is expected of a human translator, or even Google Translate for that matter, but still captured the original meaning. All also agreed that language translation is very hard.

Language use analysis - CS and FFS students collaborated on numerous, individual projects to analyzing writing in various ways. For example: a Marketing major chose an advertisement from a marketing campaign and collaborated with CS students to devise tools to compare the use of language in the English versus French versions of the same advertising copy; a dual FFS and Political Science student worked with CS students to study passages from Alexis de Tocqueville's writings of early America, looking for similarities and changes in style in pieces written at different times within the context of different historical events; and, an FFS major chose a French film and worked with CS students to programmatically analyze its use of subtitles as compared with the full script of the film. These unique and innovated projects would have been unlikely if done in either the FFS or CS courses separately, lending credence to the tremendous benefits that comes from empowering student learning through interdisciplinary projects possible using a conjoined model.

6 Conclusions & Future Work

Throughout the three successive offerings of this novel, conjoined approach to offering interdisciplinary courses our enthusiasm with this novel, conjoined approach, continued to grow. After hearing many harrowing tales of colleagues who have pursued the more traditional joined approach to interdisciplinary instruction, we found it very satisfying from a professional and personal point of view to find a technique that seems to work very well while considerably reducing the challenges typically involved. Our students and involved administrators expressed their appreciation for the approach through continued enrollment, attendance, support and commitment. The materials we have developed collaboratively have continued to form a valuable foundation as the effort continues with our next anticipated offering in Fall 2018.

A new avenue of exploration for the conjoined approach is the use of shorter-duration collaborations involving a 1-2 week collaboration between courses rather than an entire semester. For instance, we envision our existing machine translation, or a software engineering or senior design projects course offering "consulting services" with any number of non-technical courses to produce innovative software tools that could be used by those non-technical students in new and interesting ways. Ideas include creating sentiment analysis tools that could be used in courses in political science or gender studies to identify political or gender bias in writing, or author identification tools that could find misattribution of writings in history or English courses.

7 Acknowledgments

Sincere thanks and gratitude to my interdisciplinary collaborators, François Massonnat and Seth Whidden, from Villanova's Department of Romance Languages and Literature. The work reported in this paper was partially funded by NSF CPATH award IS-0829616, and by an institutional course development grant from the Villanova Institute for Teaching And Learning (VITAL).

8 References

[1] Lori Carter. "Interdisciplinary computing classes: worth the effort." SIGCSE Technical Symposium (SIGCSE 2014), pp. 445-450, 2014.

[2] Lillian Cassel, Robert E. Beck and Richard H. Austing, "Interdependence of Disciplines: Computer Science as a Full Partner," Proc. of the 5th World Conference on Computers in Education, 1990.

[3] Lillian N. Cassel and Deepak Kumar, "Distributed Expertise in Teaching," The 7th IFIP World Conference on Networking the Learner: Computers in Education, Copenhagen, Denmark, pp. 881-886, 2001.

[4] Lillian N. Cassel. "Interdisciplinary computing is the answer: now, what was the question?" ACM Inroads, vol. 2, no. 1, pp. 4-6, March 2011.

[5] Ling-Chian Changa and Greg C. Lee. "A team-teaching model for practicing project-based learning in high school: Collaboration between computer and subject teachers." Computers & Education, vol. 55, no. 3, pp. 961-969, November 2010.

[6] Debra S. Goldberg and Elizabeth K. White. "E pluribus, plurima: the synergy of interdisciplinary class groups." SIGCSE Technical Symposium (SIGCSE 2014), pp. 457-462, 2014.

[7] Jorge A. Vázquez Diosdado, et al. "Classification of behaviour in housed dairy cows using an accelerometer-based activity monitoring system." Animal Biotelemetry, 3(15), June 2015.

[8] Julie Thompson Klein. Interdisciplinarity: History, Theory, and Practice. Detroit: Wayne State University, 1990.

[9] Bill Manaris, Renee McCauley, Marian Mazzone, and William Bares. "Computing in the arts: a model curriculum." SIGCSE Technical Symposium (SIGCSE 2014), pp. 451-456, 2014.

[10] Keith J. O'Hara, Sven Anderson, David Musicant, Amber Stubbs, Thomas Way. "Team-Teaching with Colleagues in the Arts and Humanities." 49th SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2018), Baltimore Maryland, Feb. 2018.

[11] Alan L. Porter and Jan Youtie. "How interdisciplinary is nanotechnology?" Journal of Nanoparticle Research, vol. 11, no. 5, pp. 1023-1041, 2009.

[12] S. M. Pulimood and U. Wolz, "Problem Solving in Community: A Necessary Shift in CS Pedagogy," SIGCSE Technical Symposium (SIGCSE 2008), pp. 210-214, 2008.

[13] Thomas Way, Lillian Cassel, Kim Pearson, Ursula Wolz, Deborah Tatar and Steve Harrison. "A Distributed Expertise Model for Teaching Computing Across Disciplines and Institutions." The 2010 International Conference on Frontiers in Education: Computer Science and Computer Engineering, Las Vegas, Nevada, 2010.

[14] Thomas Way, Seth Whidden. "A Parallel, Conjoined Approach to Interdisciplinary Computer Science Education." Poster presentation. 21st Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2016), Arequipa, Peru, July 11-13, 2016.

[15] Thomas Way and Seth Whidden. "A Loosely-Coupled Approach to Interdisciplinary Computer Science Education." The 2014 International Conference on Frontiers in Education: Computer Science and Computer Engineering, Las Vegas, Nevada, July 21-24, 2014.

[16] Jeanette M. Wing, "Computational Thinking," Communications of the ACM, vol. 49, no. 3, pp. 33-35, March 2006.