A Loosely-Coupled Approach to Interdisciplinary Computer Science Education

Thomas Way[1], **Seth Whidden**[2]
[1] Department of Computing Sciences, Villanova University, Villanova, PA, USA
[2] Department of Romance Languages & Literature, Villanova University, Villanova, PA, USA

Abstract - *Interdisciplinary education seems to be on the lips of every university president, dean, provost, department chair, professor, student and industry professional these days. The concept of combining disciplines in a way that students will see them in the real world is appealing and valuable. Implementing a truly interdisciplinary course, where two or more disciplines are merged into a single course, can be an exciting proposition. This endeavor can be time-consuming, labor intensive, cumbersome and administratively challenging. In this paper, a less onerous approach is presented that overcomes many of the hurdles typically encountered in getting an interdisciplinary course offering on its feet. As an initial attempt, Computer Science and French disciplines were successful combined by holding two, simultaneously offered, independent courses that had numerous collaborative lessons and projects designed into them. Quantitative and qualitative results are presented, a realistic analysis of outcomes is given, and guidelines for developing a similar offering are proposed.*

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

1 Introduction

The breathtaking pace of change in computing and technology, and its widespread adoption in virtually every human endeavor, has led us to the dawn of a never before seen era of interdisciplinarity. No longer is specialization enough. Nearly all fields of human activity require an understanding and application of that field within the context of one or more other fields. Within education, interdisciplinarity is the combining of two or more disciplines into a single, cross-discipline learning experience. [7]

Beyond academia, interdisciplinarity involves the crossing of traditional topical boundaries in business, science, industry in general and across society [7]. In some cases, these new combinations are 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 [9].

There are a variety of well-recognized benefits and barriers to interdisciplinarity within the academic world, such as the siloed culture of specialization, differences between approaches to teaching and learning among disciplines, differences of opinion regarding the importance of interdisciplinarity, and a host of institutional, administrative and logistic issues. [1, 4, 7]

Traditional approaches to interdisciplinary education frequently involve a true merging of subject areas into a single course. Two, and sometimes more, faculty members collaborate to contribute content and merge material, crafting the course that presents this subject matter in a way that crosses the boundaries of the involved disciplines to produce a new and possibly never-before-offered course. Initial excitement among faculty and administrators can sometimes transform when the reality of creating and offering such a course sets in. Faculty find the effort fulfilling, but also very labor intensive, time consuming and cumbersome. Administrators find the idea compelling and promotable, but wrestle with how to fit the course into rigidly defined course categories, justify extra expenditures on faculty time and materials, and assign course teaching credit to multiple faculty who are teaching the same course, often with lower than typical enrollment.

For classification purposes, we have dubbed the collection of variations on the approaches to creating a merged-topic, high-overhead course the *tightly-coupled interdisciplinary* course approach. This name draws its inspiration from the concept of tightly-coupled parallel processing, where parallel tasks are highly dependent on each other. With a tightly-couple approach, the topics in an interdisciplinary course are highly dependent on each other as they are integrated through careful design and significant effort, to produce what is essentially a brand new discipline that contains merged elements of all original disciplines.

In this paper, we introduce the *loosely-coupled interdisciplinary* approach that is much lighter weight and overcomes many of the organizational and administrative challenges inherent with the traditional, tightly-coupled approach while still providing students with much of the same cross-disciplinary benefits that are gained with a tightly coupled course. The approach decouples much of the dependencies between the topic areas being taught, much in the way a loosely-coupled approach to parallel processing decouples individual tasks.

2 Interdisciplinary Approaches

Significant effort has been invested in a wide variety of interdisciplinary Computer Science teaching approaches [1, 4, 7, 10, 11, 13], and is an active and ongoing area of pedagogical research. These efforts have primarily focused on variations of the tightly-coupled interdisciplinary model, as most work aims at designing a single course and this is the most natural way to think about collaborating on an interdisciplinary project.

The approach of *distributed expertise* (Figure 1) is a form of collaborative education that does not necessarily require crossing disciplinary boundaries, although it provides a model that is applicable. With distributed expertise, faculty members contribute from their own backgrounds in a collaborative effort that draws proportionally from these backgrounds in support of the subject matter. It is a model that can be helpful when forming collaborations across disciplines in the early stages of designing an interdisciplinary course. [3, 11]

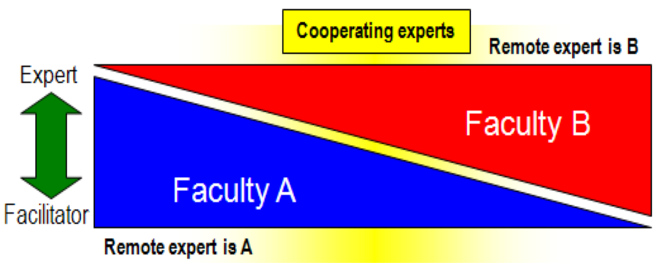

Figure 1. Digital distributed classroom expert-facilitator model continuum.

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.

While summarizing the vast breadth of experiences with tightly-coupled approaches that are reported in the literature is beyond the scope of this paper, there is great value in identifying the most successful aspects of this approach and finding ways to apply those aspects in the most efficient and effective way possible. Mining the wealth of interdisciplinary course design experience of our colleagues has been an invaluable component of designing the loosely-coupled interdisciplinary model that is presented in this paper.

3 Loosely-Coupled Approach

Given the increasing value of Computer Science to other disciplines [6], 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.

3.1 Motivation

As computing and technology use has become ubiquitous, the skill of computational thinking has become a valued and necessary intellectual commodity as a literate member of society [12]. Thus, Computer Science educators are positioned ideally to facilitate interdisciplinary courses that put computing in context [1, 2] and provide a relevant and foundational expertise that informs computational thinking across virtually all disciplines, including the arts [8].

Experience with previous efforts to create and offer interdisciplinary course models [1, 10, 11, 13] led to an understanding of many of the difficulties inherent in the traditional, tightly-coupled approaches. Thus, identifying the challenges and ways to overcome them motivated the design of a loosely-coupled interdisciplinary approach to course design.

We recognized the value of the distributed expertise approach and the strong desire at many levels of academia to provide interdisciplinary education opportunities for students and faculty. For any interdisciplinary course design approach to be reusable, it should be easily replicable and attempt to reduce the cumbersomeness that can occur with the tightly-coupled model. A goal perhaps specific to interdisciplinary Computer Sciences courses is to give students a taste of real-world problems and solutions, putting each disciplinary topic within a relevant context of the other disciplinary topics.

3.2 Model

Given the motivations we identified, the concept of the loosely-coupled interdisciplinary approach was designed. Within an interdisciplinary course, there are at least two topic areas that are combined to produce the material covered by the course. In Figures 2 and 3, these two topic areas are labeled "A" and "B". For the models shown here, these labels can indicate two distinct topics, or individual faculty members, or even two pre-existing courses.

The tightly-coupled model (Figure 2) merges topics A and B into a new topic C. Much effort and time is then invested in devising a semester-long course (indicated by the long arrow line) that thoroughly covers topic C. This diagram also illustrates the investment of time and effort by two faculty members, A and B, in preparing and presenting the topic. The diagram also shows that students are joined in attending class C, and participate together for the duration of the semester is studying topic C.

Figure 2. Tightly-coupled model.

The loosely-coupled model (Figure 3), by contrast, does not produce a brand new topic C. Instead, it remains as two distinct topics, A and B, with multiple topical “merge points” during the semester. The diagram demonstrates 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 much less cumbersome way.

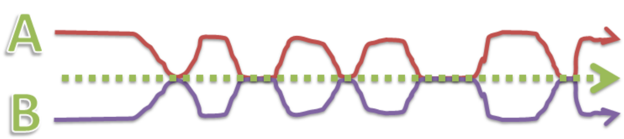

Figure 3. Loosely-coupled model.

In this new approach to interdisciplinary course design, students attend two separate classes taught by two different faculty members, but come together numerous times during a semester to join forces in learning and activities. To apply this model, two interested faculty members first recognize that there can be a meaningful collaboration between the topics of their existing courses, then identify and prepare for multiple points during a typical semester to conduct interdisciplinary activities, have their courses scheduled at the same days and times, and offer their slightly modified courses to their typical student populations.

Risk is reduced, since even if everything went wrong, two courses could run independently impact only to some of the specific material covered. Majors and minors still fulfill their requirements, and aside from disappointment that the interdisciplinary aspects were not explored, everybody is okay

3.3 Course Design

Having devised a model for a lighter-weight approach to interdisciplinary teaching, we implemented the model to combine two discipline-specific courses. To test the model, we prepared and offered two simultaneous undergraduate courses, 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 and 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 goal for this structure was to deepen and enrich the educational experiences of all involved through interdisciplinary projects, lectures, activities and discussions at the intersection of human language and its manipulation by computers. The overarching goals were to develop a portable approach for combining CS topics with disciplines that traditionally might be considered non-overlapping with or disjoint from CS; and to extend the FFS curriculum beyond its traditional confines, bringing the study of language to bear on other fields of knowledge. In addition, we hoped that the semester’s work would be consistent with the broader missions of our college and university of fostering interdisciplinary learning opportunities.

3.4 Preparation

In order to prepare for our loosely-coupled interdisciplinary teaching experiment, we met a number of times during the summer leading up to the fall semester in which the courses were to be offered. The outcomes of these meetings, which continued throughout the semester in which the courses were offered, included:

- Simultaneous course scheduling - arranging with the university registrar to schedule our courses on the same days and times in nearby classrooms. At least one classroom should be sufficient large to accommodate meetings of all students from both courses.
- Regular solo course preparation - we continued, as in past offerings, to independently prepare our course to assure we covered the necessary material for our major and minor students.
- Planning periodic join-points - we identified approximately 10 topics that our combined group of students would benefit from joint learning, including lectures, activities, discussions and projects.
- Brainstorming student interactions - we invested quite a bit of time in coming up with relevant and engaging ways to have our students interact, usually through

classroom activities, discussions and interdisciplinary projects that required expertise from both disciplines.

- Imagining student projects - Early on, we envisioned that projects would be the key to maximizing student collaboration while maintaining focus on disciplinary knowledge. Software tool-based projects that could analyze French texts of various forms, and that interdisciplinary student teams could create, held promise.

3.5 Implementation

Implementation was done for the Fall 2012 semester and was relatively straightforward, involving an important initial joint class meeting following by both independent, parallel course single-discipline meetings and joint, interdisciplinary meetings. The CS and FFS courses met separately for the first meeting of the semester so that instructors could prepare students for the semester to come. In the next course meeting time, CS and FFS students met together in the same classroom where a joint presentation by the two faculty members was made to introduce the students to the interdisciplinary aspects of the course.

We embraced one notable outcome of previous work with interdisciplinary distributed expertise that found that it was vitally important to have students from different classes who will be collaborating meet each other very early in the semester [11]. Our experience in this offering was that it made a significant difference in how the semester went as students all felt like they knew each other and there was no resistance to collaboration and no academic tribalism as can be seen in approaches that resemble the loosely-coupled concept [11].

Throughout the semester, the courses continued to run independently in regards to the domain-specific content for each course while meeting jointly 10 times for interdisciplinary lectures, discussions and activities.

As the semester progressed, FFS students were required to design a culminating literature analysis project that CS students were required to collaborate on. Interdisciplinary teams of students were formed, with the CS students acting as technical consults to the domain expert FFS students. A number of joint, whole-class brainstorming sessions were held where FFS students presented ideas, CS students proposed solutions and identified aspects that were either too challenging to be accomplished in a single semester or otherwise unworkable, and faculty moderated and guided the process as needed.

Next, CS students invested time designing and implementing tools that sometimes included use of available online tools, reuse of pre-existing solutions being emphasized as an essential part of the process. These solutions were evaluated by FFS students a couple times during the process to provide informal feedback.

During the final two weeks of the course, students met jointly as teams to actively collaborate on using tools that were created by the CS students, modify the tools to produce better results, and generally to interact in a very real-world, interdisciplinary way.

4 Evaluation

The effectiveness of this approach, and the two courses in which we implemented the approach, 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.

4.1 Student Performance

For the CS course, the average semester grade for the seven students enrolled in the course was A-, with students earning four As, two A-s and one B. All assignments were completed by all students, and the variability in grading was due to thoroughness and consistency of the work done.

The CS course was project-oriented so rather than a final exam students were evaluated on completion of a final project that included significant team-based participation both internally to the CS course and with students in the FFS course. There was significant enthusiasm for the course activities, particularly when they involved the cross-course project work that was done.

For the FFS course, the average semester grade for the eighteen students was B+, with students earning two As, four A-s, six B+s, five Bs and one B-. 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 without our CS partners, and the degree of effort evident in their final project, which was a presentation (in French) of the individual project that each FFS student undertook.

The final papers indicated, in most cases, a high degree of creativity and enthusiasm; the students appreciated the ability to pursue individual projects that were an extension of their existing interests. Every student demonstrated throughout the semester the extent to which this new approach to working with language using computers made them look at familiar materials and an entire field of study in a brand new way. In addition, they 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 1 shows the average values for the four categories measured.

Table 1. Average values per class for evaluation instrument measure taken at start and end of Fall 2012 semester.

	CSC 5930		**FRE 1140**	
Categories (0 is lowest, 9 is highest)	**start**	**end**	**start**	**end**
1. Skill level with computers	6.9	8.0	5.0	5.7
2. Computer programming ability	6.1	6.7	1.3	1.3
3. Familiarity with Writing and Stylistics in French	0.3	2.6	5.0	6.9
4. Uses of CS in Writing and Stylistics in French	1.7	5.7	2.8	6.8

Improvement was seen in almost all areas measured (Table 2). Students in the CS course showed slight but measurable improvement in computer skill (1.1) and programming ability (0.6), which was expected and encouraging given that all students were at the junior or senior level and had relatively strong skills at the outset. CS students improved significantly (+2.3) in their familiarity with the subject of the FFS course in spite on only having had a few formal, in class exposures to the topic. This improvement is likely due to the organic exposure to the topic in the course of applying their CS knowledge to solving problems for students in the FFS course, which also led to a marked increase in familiarity (+4.0) with how CS can be used in the study of Writing and Stylistics in French.

One phenomenon that we discovered was in the FFS students’ assessments of their own knowledge of and facility with computers. The short-answer follow-up questions to the numerical indications make clear the fact that the relatively small gains in these two categories reflect not a lack of learning on behalf of the FFS, but rather an inflated sense of what their true abilities were, before they had the opportunity to see the CS students really use computers. This interpretation is supported, we believe, by the increase in familiarity in how CS can be applied to the study of French texts (+4.0). While CS students appeared to have a more realistic view of their own technical capabilities at the outset, there was significant benefit to them gained from applying their science background to solving real and challenging problems from a non-computing discipline.

Table 2. Improvement in evaluation instrument measures per class and combined taken at start and end of Fall 2012 semester.

	Improvement		
Categories (0 is lowest, 9 is highest)	**CSC 5930**	**FRE 1140**	**Overall**
1. Skill level with computers	+1.1	+0.7	+0.9
2. Computer programming ability	+0.6	0.0	+0.3
3. Familiarity with Writing and Stylistics in French	+2.3	+1.9	+2.1
4. Uses of CS in Writing and Stylistics in French	+4.0	+4.0	+4.0

In the qualitative written responses, CS students demonstrated noticeable improvement in understanding of the subject matter of and uses of CS ideas for the Writing and Stylistics in French. These students observed that the experience demonstrated “knowing the limitations of computers can help students understand when or how tools are useful and when they are not” and that there is value in interdisciplinary collaboration because “programming can solve problems in areas that are much different from computer science.” The CS students were particularly satisfied with being able to devise and implement complex software tools that used their programming skills and creativity to assist students in a French course in ways they never imagined prior to this experience.

In their responses to the same questions, FFS students were able to articulate much more clearly how CS can be of significant help in learning about French written texts and their related issues of style. In addition, after seeing French Writing and Stylistics from the CS perspective, students spoke of now being able to see “many things about style and grammar that sometimes go unnoticed.” In addition, they developed an appreciation for (at least some of) “the complexities that come with computer translation programs.” Finally, FFS students felt that each discipline enhanced the other, saying that Machine Translation can help “fully capture the intricacies of a work of literature,” and that “understanding stylistics gives more options and ideas about ways to analyze a text using machines.”

4.3 Observations

Throughout the process of creating and conducting this interdisciplinary endeavor, 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.

Students and faculty alike gained a deeper understanding of the importance, in FFS, and other language-oriented disciplines, of keeping an open mind to what constitutes a "text" worthy of study. Indeed, as more and more FFS students are pursuing majors and minors in other fields of study, it becomes increasingly incumbent on FFS as a discipline to think beyond the traditional definitions of literature, in courses when it is appropriate. In FRE 1140, for example, since students were allowed to pursue a project of their own design, there was a wide array of "texts" that they chose, each one worthy of study; a Marketing major chose an advertisement from a marketing campaign; a dual FFS and Political Science student studied passages from Alexis de Tocqueville's study of early America; and one FFS major chose a French film and its use of subtitles. None of these approaches would have been likely in a traditional Writing and Stylistics course, and so one of the tremendous benefits to learning came from empowering the students to suggest texts to work on.

5 Guidelines

The loosely-coupled interdisciplinary approach enabled us to offer a lightweight, easily implemented interdisciplinary pair of courses. We believe that replicating this approach, while taking a concerted effort, can be done with far fewer of the challenges typically seen with tightly-coupled interdisciplinary approaches. The following checklist of steps to follow comprises our recommended roadmap for the design of a loosely-coupled interdisciplinary course duo that matches Computer Science with another discipline.

For junior faculty eager to apply the latest ideas in interdisciplinarity yet wary of straying from their specialization, we believe this loosely-coupled approach to be ideal. By following this roadmap, a junior faculty's tenure progress can be protected since their own course in the pair is officially taught is within their own discipline. As a bonus, doing so can simultaneously fulfill departmental, college and university interdisciplinary initiatives or missions.

5.1 Steps to Implement

While there are likely to be as many ways to implement a loosely-couple interdisciplinary model as there are faculty members, the following items summarize what we view as essential steps in this process.

- ✓ Identify a colleague with whom you have a good rapport. You will spend a lot of time collaborating both outside and inside the classroom, so your styles of preparation and teaching should mesh well.
- ✓ Identify courses you each teach that could combine well in a loosely-coupled approach. Note aspects of the non-CS topic that lend themselves to a computational solution of some sort.
- ✓ Determine if there are enough common topic points that your two disciplines. These points are the truly interdisciplinary components of the semester where students from both courses will meet to learn and collaborate.
- ✓ Meet and sketch out an initial plan for how a semester would flow. Organize topics and meetings within the semester syllabus and schedule to accommodate the loosely-coupled approach.
- ✓ Seek approval of necessary administrators at the departmental, program, college and university levels. Once you have a plan that works for a course pair that you believe would be a valuable contribution to your institution's offerings, get the needed approvals.
- ✓ Arrange to have courses offered at the same days and times nearby each other. The registrar's office is an invaluable ally in making the loosely-couple model workable.
- ✓ Promote the collaborative course offerings, encourage students to enroll. Get the word out to whoever is advising students who are eligible for your courses, and take advantage of any opportunities available to promote your unique offering.
- ✓ Schedule topics to be taught separately and together, coordinate interdisciplinary activities. Once it really is going to happen, spend more time planning out each week of the semester and all of the separate and combined lectures, activities, assignments and projects that will form the interdisciplinary aspects of the courses.
- ✓ Prepare materials, discussion starters, and activities. Getting out ahead of the schedule as much as possible will enable the more involved synchronization needed versus a typical, single-subject course.
- ✓ Offer the courses and be prepared for change and adaptation. It is difficult to anticipate exactly how such a unique approach to combining two courses will proceed, so anticipating the need to modify plans on the fly in response to progress will be helpful.

6 Conclusions & Future Work

The design, preparation and offering of this new loosely-coupled approach to offering interdisciplinary courses were very satisfying from a professional and personal point of view. Students were regular and enthusiastic attendees and participants in a wide variety of learning activities, some of which were quite different from activities they had experienced in other courses. Materials were developed and collected so that the course can be offered in the future with a reduced quantity of preparation.

Student output included producing software specifications and usable software tools of potential interest to others in our university community who might work with language analysis, including languages other than French. Learning was significant, as demonstrated by the evaluation instrument, particularly in the area of the interplay between the application of Computer Science technology to the study of Writing and Stylistics in French, which was a primary goal of the course. We feel that this unique combined course offering has demonstrated that a loosely-coupled approach to interdisciplinary teaching can challenge students, facilitate quicker development and offering of such interdisciplinary opportunities for faculty and students, and ultimately lead to greater, more enjoyable, and more relevant learning than a more traditional approach.

We are offering the pairing a second time in the Fall 2014 semester, which will provide an opportunity to compare and improve the approach. The loosely-coupled approach is also being discussed for proposed future course pairings of CS and Criminal Justice (Cybersecurity), CS and History (Big Data Analysis of the Historical Record), CS and Political Science (Analysis of Political Bias in the Media), and CS with English or Spanish in a similar, computational approach to literary analysis. The loosely-coupled approach is being redefined more specifically as a "conjoined courses" approach to more precisely describe the nature of the collaboration, with future evaluation and exploration of the approach expected.

7 Acknowledgments

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). Thanks to John Kelley and Carol Weiss for the sage advice and guidance in the development, implementation and evaluation of this work.

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

[8] 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.

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

[10] 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.

[11] 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 (FECS 2010), Las Vegas, Nevada, July 12-15, 2010.

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

[13] Ursula Wolz, C. Ault and T. M. Nakra. "Teaching Game Design through Cross-Disciplinary Content and Individualized Student Deliverables," The Journal of Game Development, 2(1), Feb. 2007.