TUES Type 2 Proposal – Interdisciplinary

Computational Thinking through Computing and Music

What Problem Are We Trying to Solve?

Computational thinking (CT) has been an important component of educational research and development projects in computing ever since Jeannette Wing coined the term in 2006 [44, 45]. Wing defines CT as “solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science” and “using heuristic reasoning to discover a solution.” Widespread agreement that the ability to think computationally is an important characteristic of an educated populace has led to numerous efforts to engage students in CT from grade school through college [2]. A common characteristic of successful efforts (as judged by number and quality of publications) is the presence of a *context* to which students can relate and within which CT is presented [12, 22].

The problem, then, is to find a context that engages learners while providing a framework that is deep and rich enough to support CT. Our recently completed CPATH project (see page 4) demonstrated that *music* is just such a context [10, 13, 35, 36]. We now propose to build on that research by developing and evaluating more in-depth learning activities and corresponding assessments that focus on CT by combining computing and music. We further propose to develop assessments that “operationalize” what it means for a student to possess CT skills.

What Do Computing and Music Have To Do With Each Other?

Computing and music share deep structural similarities. For starters, both rely on notational symbol systems. Programming loops are typically delineated with opening and closing curly brackets { }, parentheses, or levels of indentation. Music loops are delineated with begin and end repeat signs {} or initiated by “D.S.” (Italian: *dal segno*), which instructs musicians to “repeat back to the sign,” typically designated as 𝄋. As in programming, musical iteration can also make use of loop control variables. For example, Figure 1 shows a loop in which the music changes the second time through.

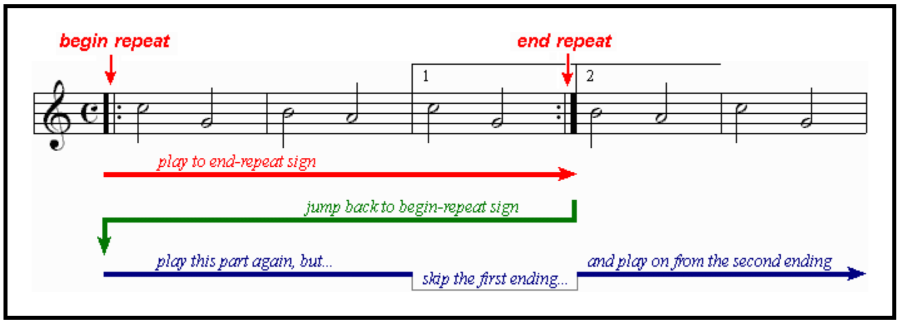


Figure 1. Musical iteration with a loop control variable. [37]

Both computing and music have logic and flow. Figure 2 shows the logic one student saw in The Beatles’ *All You Need Is Love.* If one were to turn this flowchart into a computer program, it would not only contains loops, but `if` and `switch` statements as well.

One can also go the other way, converting musical concepts into computer programs. For example, the Scratch [26, 31] program in Figure 3a plays Jimmy Page’s famous guitar riff from Led Zeppelin’s *Kashmir*. This code works properly, but consider the many CT concepts learned by transforming the code in Figure 3a to 3b, which plays exactly the same riff.

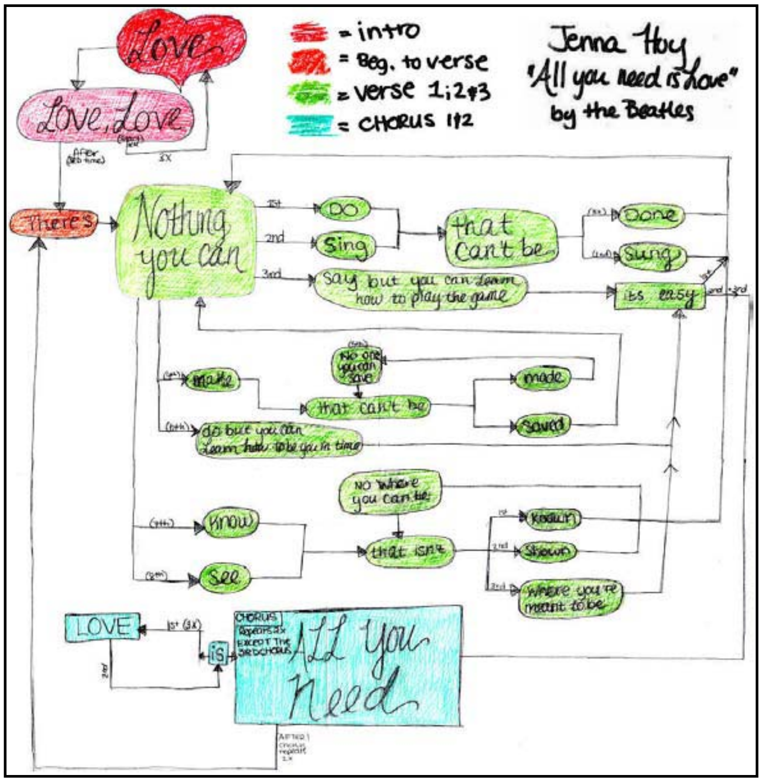
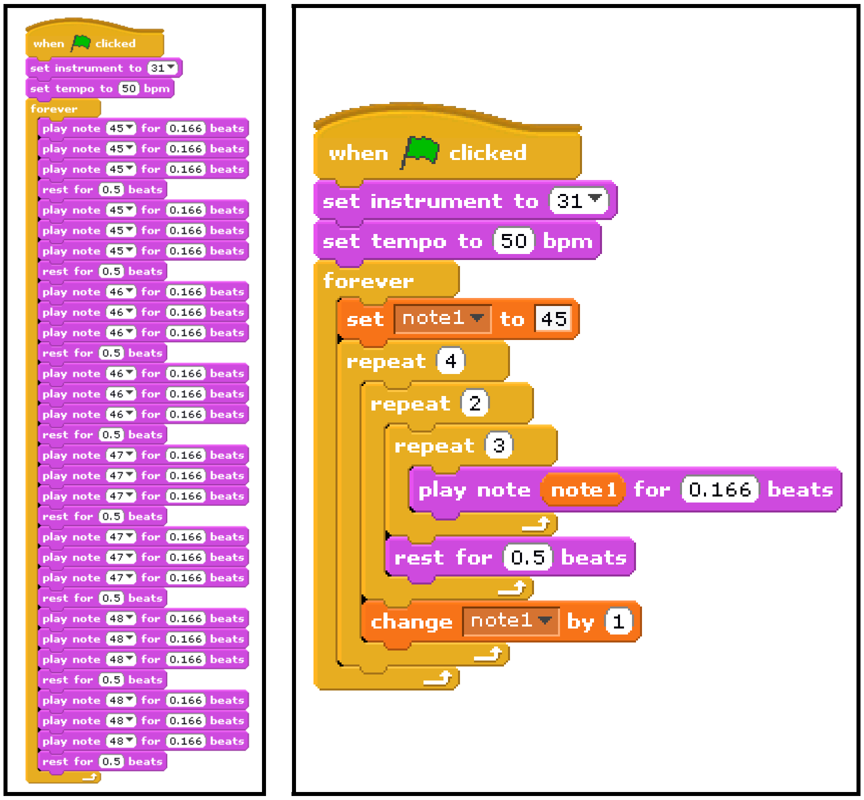


Figure 2. A song flowchart. [19]



3a

```
when [green flag] clicked
set instrument to 31
set tempo to 50 bpm
forever
    play note 45 for 0.166 beats
    play note 45 for 0.166 beats
    play note 45 for 0.166 beats
    rest for 0.5 beats
    play note 45 for 0.166 beats
    play note 45 for 0.166 beats
    play note 45 for 0.166 beats
    rest for 0.5 beats
    play note 46 for 0.166 beats
    play note 46 for 0.166 beats
    play note 46 for 0.166 beats
    rest for 0.5 beats
    play note 46 for 0.166 beats
    play note 46 for 0.166 beats
    play note 46 for 0.166 beats
    rest for 0.5 beats
    play note 47 for 0.166 beats
    play note 47 for 0.166 beats
    play note 47 for 0.166 beats
    rest for 0.5 beats
    play note 47 for 0.166 beats
    play note 47 for 0.166 beats
    play note 47 for 0.166 beats
    rest for 0.5 beats
    play note 48 for 0.166 beats
    play note 48 for 0.166 beats
    play note 48 for 0.166 beats
    rest for 0.5 beats
    play note 48 for 0.166 beats
    play note 48 for 0.166 beats
    play note 48 for 0.166 beats
    rest for 0.5 beats
```

3b

```
when [green flag] clicked
set instrument to 31
set tempo to 50 bpm
forever
    set note1 to 45
    repeat 4
        repeat 2
            repeat 3
                play note note1 for 0.166 beats
            rest for 0.5 beats
        change note1 by 1
```

Figure 3. Two versions of Jimmy Page's *Kashmir* riff programmed in Scratch. [33]

List and array data structures can be used to represent pitches and durations. Figure 4 shows an array (or indexed list) of MIDI note values paired with an array of note durations (in fractions of beats) that plays part of *Row, Row, Row Your Boat.* Using such structures, one can explore synchronization when the values are read by multiple threads with entrances staggered in time, resulting in the performance of a canon (or round).

```
when [flag] clicked
broadcast Initialize and wait
set ListIndex to 1
repeat until ListIndex > length of Notes
    play note item ListIndex of Notes for item ListIndex of Rhythms beats
    change ListIndex by 1

when I receive Initialize
set Clarinet to 72
set instrument to Clarinet
set tempo to 120 bpm
```

we must use a "repeat until" loop with our own loop index (as opposed to a "repeat n" loop) so that we have access to the loop index to use to access individual items in the list

	Notes		Rhythms
1	55	1	1
2	55	2	1
3	55	3	0.67
4	57	4	0.33
5	59	5	1
6	59	6	0.67
7	57	7	0.33
8	59	8	0.67
9	60	9	0.33
10	62	10	2
11	67	11	0.33
12	67	12	0.33
13	67	13	0.34
14	62	14	0.33
15	62	15	0.33
length: 27		length: 27	

4a 4b

Figure 4. Processing Scratch lists of notes and rhythms for *Row, Row, Row Your Boat*. [35]

Why Is the Relationship Between Computing and Music Important to Transforming Undergraduate Education in STEM?

The examples above show that fundamental concepts in music can be both expressed and taught using the language of computing and vice versa. Our work builds on this relationship to teach STEM concepts to undergraduates in ***all*** disciplines through interdisciplinary General Education (GenEd) courses at the introductory level and discipline-specific music and CS courses at more advanced levels.

The interdisciplinary component of our approach is key to its success. We do not use music simply as a “hook” or consider music in any way subservient to computing. On the contrary, in some activities computing is a means to achieve the desired end: music. That is, students may write and interact with programs to “play” computers as if they were musical instruments à la the “laptop orchestras” that are now commonly found on many university campuses [5, 39, 43], including our own. Our activities not only have students use loops and conditionals to sequence prerecorded sounds and write algorithms to generate new MIDI sounds, they also have students write and manipulate code in real time to control a program’s execution “on the fly,” an increasingly popular technique used in actual concerts where the artists practice “live coding” [4, 42]. Such dynamic activities require considerable feel for the musical nature of the project in addition to the underlying CT and coding techniques. One must clearly have achieved some level of CT competency to perform a piece using live coding.

Thus, we attempt to help students develop CT skills through computing and music in a way that can provide STEM education to all students, regardless of their major or prior experience with either discipline.

Where Do These Ideas Come From?

Prior NSF Support

This proposal builds on our work with *Performamatics*, an interdisciplinary course and teaching strategy development effort funded by NSF award 0722161, "CPATH CB: Performamatics: Connecting Computer Science to the Performing, Fine, and Design Arts" ($380,162, July 2007 to June 2010) [14, 23]. Jesse Heines was PI and Gena Greher and Fred Martin were co-PIs, with Sarah Kuhn as Senior Personnel. Alex Ruthmann joined the faculty in 2008 and quickly became a significant contributor to these efforts.

Courses Developed

We developed two course models, or teaching strategies, as we implemented the Performamatics approach:

- *synchronized*, in which a joint project is completed collaboratively by students in two upper-level courses in separate disciplines, and
- *hybrid*, in which a single course is taught by two professors from two disciplines, simultaneously teaching together in the same classroom.

The first model allowed students from specific majors to delve more deeply into context-based activities, while the second provided opportunities to actually study another discipline with the help of students from other majors. The latter model resulted in two new GenEd courses that have seen increasing enrollments in each subsequent semester:

- *Tangible Interaction Design* [25], which focuses on understanding how people interact with designed objects in the world, and
- *Sound Thinking* [15], which explores the intersection of music and technology through the art and science of digital audio.

Papers on these courses were presented at the ACM Creativity & Cognition conference [24] and at conferences of the ACM SIGCSE [16, 36], the Association for Technology in Music Instruction [10, 34] and the Consortium for Computing Sciences in Colleges (Northeast Region) [23]. Oxford University Press has recently awarded a contract to Greher and Heines to write a book on interdisciplinary teaching entitled *Computational Thinking through Sound: Teaching the Art and Science of Music Technology*. A refereed workshop has been presented at one conference [35] and accepted for presentation at another [17]. In addition, the project generated invitations to participate in two refereed conference panels [13, 40].

Impact on Students

The hybrid courses we developed were brand new. We had no historical data to compare to, and the courses did not replace or serve as alternatives to any existing courses. Therefore, we did extensive student surveys to assess the effectiveness of the courses and their impact on student attitudes and perceptions. The data in Table 1 (next page) show representative results of those surveys. They indicate the percentage of students in the 2009 and 2010 offerings of our *Sound Thinking* course (a combined total of 27 students) who responded either "Strongly Agree" or "Somewhat Agree" to the corresponding statements. These results support our observations that both science and arts students increased their knowledge of one another's fields and improved their abilities to communicate and collaborate with peers in another discipline. Further, we analyzed classroom projects to assess the depth, nature, and development of students' CT.

Impact on Faculty

A key component to offering more engaging undergraduate STEM courses is revitalizing the faculty. Engaged, excited, committed faculty members are clearly more effective teachers

Table 1. Percent of students (n=27) in two sections of *Sound Thinking* who responded either Strongly Agree or Somewhat Agree to specific statements.

This course and activities helped me better understand	
Computer Science.	70%
the Arts.	77%
My experience in this course increased my interest in	
Computer Science.	48%
the Arts.	59%
collaborating with others.	65%
Compared with other courses, this course was more	
engaging.	63%
educational.	44%
time consuming.	41%
real world.	48%
useful to my future.	48%
difficult.	26%
boring.	22%
Compared with other courses, this course	
increased my confidence in my abilities.	63%
discouraged me from doing anything with computing in the future.	11%
made me want to recommend this course to others.	70%
made me more creative.	59%
increased my ability to present/perform my work.	56%
improved my communication skills.	56%
increased my ability to collaborate with others.	74%
increased my interest in the design process.	78%

than those who are tired, uninterested, and grudging. Performamatics allowed participating CS faculty to become more engaged with the arts and with arts students and to see their discipline through a different lens.

One CS faculty member already had experience blending the arts and computing, but the other did not. The second was an amateur musician, and the project allowed him to bring his music background to his professional setting, opening up many new opportunities for teaching, collaboration, and publication, including the Oxford University Press book mentioned earlier. This professor significantly changed his approach to teaching in some situations, assigning more open-ended projects, a change well received by students.

Thus, both CS and music faculty showed changes in pedagogical approach and affective changes toward the experience of teaching. Such transformations are an essential but often overlooked element of institutional and curricular change.

How Will We Achieve Our Objectives?

Our overall goal is to *develop and disseminate ways to enhance students' grasp of computational thinking by engaging them in fundamental concepts that unite computing and* *music*. This goal will be achieved by performing the tasks related to the three component objectives discussed below.

Objective 1: Develop new learning activities that integrate computing and music.

As mentioned in the Project Summary, our existing materials teach concepts such as

- modularization, by breaking songs down into their components;
- looping and subroutines, by noting where musical phrases are repeated intact and with small variations (requiring parameters);
- logic flow, by creating musical flowcharts; and
- algorithm development, by writing programs that generate music and manipulating those programs via "live coding."

These concepts and activities have worked well for a broad range of students taking their first GenEd music or their first GenEd technology course. To meet the TUES goal "to improve the quality of STEM education for all undergraduate students," we propose to expand these activities considerably in our GenEd offerings and into upper-level music and CS courses.

New activities will teach more advanced computing concepts or go into more depth on topics covered in our existing materials. They will require the concepts in our existing course as a prerequisite. Such concepts will include

- threads and synchronization, taught by writing programs that play multiple parts simultaneously;
- web development, by creating web pages that incorporate music in a variety of ways (actual MIDI sound generation as well as prerecorded sounds) and in a variety of scenarios (musical introductions, as dynamic enhancements to games, as screen readers, etc.);
- the use of APIs, by tapping the capabilities of multiple tools to do more than is possible by any one alone; and
- data structures, by cataloging sounds and creating viewers to browse them in a variety of ways (by artist, genre, time, source, etc.).

These concepts may be covered in complementary, stand-alone courses that require *Sound Thinking I* as a prerequisite, or they may be covered in music courses that are paired ("synchronized," i.e., taken concurrently) with existing mid- to upper-level CS courses, such as *Software Engineering* or *Graphical User Interface Programming*. Alternatively, modules that cover these more advanced concepts may be integrated into appropriate courses to take advantage of the power of music to add relevance to the teaching of those concepts.

The **outcome** of these efforts will be a set of models, course modules, and frameworks that span the undergraduate years. Course materials will include lecture notes, class activities, code examples, and homework assignments. These will be posted on a website from which they can be freely downloaded, as well as contributed to other online databases such as Ensemble, the NSF-sponsored distributed digital library for computing education [28]. They will also be the main focus of the materials that we work with and distribute to participants in our workshops.

We will **evaluate** these materials in two ways. First, the materials will obviously be used in our own the classes, and in those we will administer formal quizzes to assess student learning. Second, we will engage as reviewers faculty at other universities who, like us, are working at the intersection of computing and music. Faculty who have already agreed to do so are:

- Jeanne Bamberger, Prof., Dept. of Music, MIT (emerita) and U.C. Berkeley (visiting)
- Judith Bowman, Prof., Music Education & Music Technology, Duquesne Univ.
- Rick Dammers, Asst. Prof., Music Education, Rowan Univ.
- John Maloney, Staff Researcher, Lifelong Kindergarten Group, MIT Media Laboratory

- Bill Manaris, Prof., Dept. of Computer Science, College of Charleston
- Jeff Popyack, Assoc. Prof., Dept. of Computer Science, Drexel Univ.
- Robert Rowe, Vice Chair, Dept. of Music & Performing Arts Professions, NYU
- Kimberly Walls, Prof., Music Education, Auburn Univ.
- Peter Webster, John Beattie Prof. of Music Education, Northwestern Univ.
- Ursula Wolz, Assoc. Prof., Dept. of Computer Science, The College of New Jersey

Letters of support from these individuals are included as supplementary documents.

Objective 2: Devise, test, and evaluate new techniques for assessing students' CT knowledge.

Our prior research showed that it is challenging to assess student achievement and the impact of project-based learning on developing students' grasp of CT, particularly in an interdisciplinary, constructivist environment. We have therefore promoted "operationalizing" student learning of CT to a primary objective in this proposal. Drawing on the National Academies report on CT [27], we will develop a working definition of CT in the context of music. We will then work to devise and test ways to assess CT through experiences that challenge students to explore and develop CT expertise and demonstrate that expertise through applied sound and music projects.

One assessment technique we will employ is "think-aloud" music commentaries [6, 41]. This technique requires students to verbalize what they are thinking out loud as they wrestle with a problem or engage in a creative activity, thus providing a window into their thought processes [9, 32]. The technique has proven valuable in computer interface usability studies [29], where it lets researchers in on what users are thinking as they work through assigned tasks. We will have students record themselves as they verbalize their thoughts and explain their work to others. Recordings will then be analyzed for patterns that reveal evidence of CT [18]. We will categorize these patterns and transform them into rubrics, thus allowing us to apply them consistently over a range of student projects.

Another is the "create-play-share-reflect" cycle, which closely mirrors musicians' creative processes. Resnick [30] describes this cycle as "a spiraling process in which children **imagine** what they want to do, **create** a project based on their ideas, **play** with their creations, **share** their ideas and creations with others, **reflect** on their experiences — all of which leads them to **imagine** new ideas and new projects" [*boldface in original*]. We have used an analogous cycle in our *Performamatics* courses. In the context of *Sound Thinking*, we will video students as they work together in teams and try out their ideas. As with the "think-aloud" technique, we will then analyze these videos for evidence of developing CT and create rubrics for quantifying that development.

Finally, we will create a number of exercises that model CT processes in music, mimicking the concept of an "étude," i.e., "a complete and musically intelligible composition exploring a particular technical problem in an esthetically satisfying manner" [8]. Our "compositions" will be assignments positioned at the intersection of computing and music, such as writing a single program that plays a variety of scales (major, harmonic minor, pentatonic, etc.) given a list of intervals (2 semitones, 6 semitones, etc.) between the notes of the scale. All student groups will get the same assignment, and they will be asked to explore creative solutions. Once again, we will create rubrics that capture various characteristics of the students' solutions, including creativity and code efficiency.

The **outcome** of these efforts will be a set of approaches to measuring and assessing CT gains in interdisciplinary, project-based courses with examples drawn from the musical computing domain. These will be documented in a manner that allows their overall approach to be transferred to other paired domains, thereby significantly broadening their impact. As with Objective 1, sample exercises and assessment approaches will be posted on a free website, and we will share materials and lessons learned in the workshops we offer (see Objective 3 below). We also expect that this research will result in a number of publications and conference papers, further contributing to their dissemination.

To **evaluate** our work on this objective, we will use at least two researchers who will independently code the videos of students' "think-aloud" design sessions and assess performance on student "études." We will quantify inter-rater reliability (the degree of agreement or disagreement of researchers coding independently) to test how clear and how consistently applicable our rubrics are.

Objective 3: Disseminate modules and rubrics via workshops for 4- and 2-year college faculty to build expertise and foster interdisciplinary collaboration between computing and music.

Our primary method of dissemination is by hosting workshops with faculty colleagues. Our workshops combine three strategies, which we believe will be especially fruitful.

First, our workshops will disseminate our **model of collaboration** as well as the materials we have developed. Exportable aspects of that model include

- the process of planning to build synergies between the disciplines,
- adding aesthetics to otherwise technical studies,
- co-teaching to strengthen students' abilities to recognize intersections between disciplines,
- maintaining each field's individual integrity so that one is not "in service" to the other, and
- using meetings at conferences to collaborate under the mentorship of others.

Interdisciplinary collaboration has generated many new subfields, with computing and music being but one. We expect our model of collaboration to apply to other interdisciplinary efforts, just as our own was inspired by work in Artbotics, an intersection of art and robotics [47].

Second, we will require workshop participants to **attend in interdisciplinary pairs**, preferably from the same institution. This will ensure that the workshop itself models interdisciplinary collaboration, is highly relevant to STEM, and produces outcomes that connect directly to the participants' own institutions and situations. Paired attendance will also increase the probability that workshop outcomes are sustainable, that is, that participants will follow through and explore opportunities for interdisciplinary collaboration when they return to their home institutions. As workshop leaders, we will also model interdisciplinary collaboration in the process of presenting the workshop content.

Third, all participants will **produce actual course documents and materials** for their own interdisciplinary collaborations during the workshops themselves. We realize that participants may not be able to actually co-teach at their universities due to a variety of factors, including campus logistics, scheduling, teaching load requirements, and cultural inertia. However, if they work with one of their own colleagues in the workshop and produce preliminary course materials that are relevant to a potential collaboration "back home," the probability that they will at least incorporate interdisciplinary activities into their own existing courses will be maximized.

Workshop **outcomes** will include actual syllabi, interdisciplinary course modules, learning activities, and evaluations to be used at the participants' own institutions. Where allowed, these will be shared online. We will **evaluate** these outcomes by having participants complete an evaluation immediately following the workshop experience. In addition, participants will be asked to complete a modestly revised evaluation six months to one year after the workshop to assess the longer-term impact of their interdisciplinary efforts. An important component of this follow-up evaluation will be a request for documentary evidence of the integration of interdisciplinary elements into their courses, e.g., syllabi, project instructions, examples of student work, etc. These assessments will involve studying materials from existing courses to which interdisciplinary elements that support CT have been added as well as new courses designed specifically to encourage CT.

What Is Each Team Member's Role?

Table 2 identifies the primary and secondary roles that each team member will play in the project, although in reality all team members will probably contribute to all aspects of the project to at least some degree. Details of each person's role related to the project objectives are provided in the subsections that follow the table.

Table 2. Team members' roles.

	Heines *PI*	Greher *Co-PI*	Ruthmann *Co-PI*	Kuhn *Sr. P.*	Martin *Sr. P.*	Lipscomb *Cons.*	Grad. Student
design teaching materials	🟢	🟢	🟢	🟠	🟠		🟢
implement teaching materials	🟢	🟢	🟢				
review teaching materials				🟢	🟢	🟢	
teach	🟢	🟢	🟢				
design assessments	🟢	🟢	🟢	🟠	🟠	🟢	
test assessments	🟢	🟢	🟢			🟠	
evaluate assessments	🟠	🟠	🟠	🟢		🟢	
design website	🟢		🟢				🟢
implement website	🟢						🟢
lead workshops	🟢	🟢	🟢				🟠
evaluate workshops				🟠	🟠	🟢	
administer project	🟢						🟠
evaluate project	🟢			🟠		🟢	🟠

Legend: 🟢 *primary role*
🟠 *secondary role*

Objective 1: Develop new learning activities that integrate computing and music.

Heines, Greher, and Ruthmann have been collaborating for three years on the existing *Sound Thinking* course. This trio developed the learning activities described earlier, and they will be the main developers of the new ones to be created. They will be assisted by Martin and Kuhn, who were integral parts of the Performamatics project and thus have a deep understanding of interdisciplinary approach we are embracing. Thus, while Greher, Heines, and Ruthmann are the primary instructors on this project, all five professors will be members of the creative team that designs the courses and crafts new learning activities.

Development of new materials will begin in earnest at the outset of the project (requested for the Fall 2011 semester), when Heines and Ruthmann will be on separate sabbatical leaves. Heines will be working with

- Jason Freeman of the Georgia Tech School of Music, who attempts to create "new connections among composers, performers, and listeners, using extended notation and new technology to experiment with the ways in which music is created, performed, and heard;"
- Mark Guzdial of the Georgia Tech College of Computing, the main force behind "learning CS through a media computation context;" and
- John Maloney of the MIT Media Laboratory Lifelong Kindergarten Group, the lead programmer for Scratch.

Ruthmann will be working with

- Andrew Brown of the Queensland Conservatorium, a leading practitioner of musical live coding, using computing as a platform for creative musical expression and learning.

Letters of invitation from Freeman, Guzdial, Mitchel Resnick (Director of the Lifelong Kindergarten Group), and Brown are included with this proposal as supplementary documents.

The purposes of working with these other researchers are (1) to gain a broader perspective on the intersection of computing and music and (2) to observe first-hand the learning activities they use to engage learners. We anticipate that many of these activities can be adapted for use either in the existing *Sound Thinking* course or in upper-level courses that explore the intersection of computing and music. Those adaptions will be made through discussions with the entire creative team. Materials that result from this effort will be reviewed by the external reviewers identified in the previous section, revised per their recommendations, and then tried out in the Spring 2012 offering of *Sound Thinking* and other CS and music courses taught by members of the project team. They will be refined again during summer of 2012 in preparation for use in more advanced courses during the Fall 2012 semester.

These tasks are shown in the Project Timeline at the end of this major section.

Objective 2: Devise, test, and evaluate new techniques for assessing students' CT knowledge.

As noted earlier, we anticipate that this will constitute the most challenging objective of this project. By the same token, it has the greatest potential for the broadest impact. We will strive to develop techniques that are discipline-independent or can be transferred to other disciplines with minimal modification. These efforts will provide tools for the evaluation of other interdisciplinary efforts as well as our own. This research should also influence future efforts in the important pursuit of techniques that operationalize CT.

Development of these techniques will involve an iterative process. The creative team will begin by identifying student skills that demonstrate CT and then devise activities that both develop and assess those skills. The team will create music-based assessments, quizzes, and tests that measure students' mastery of the identified skills and assess the validity of those tests by administering them in classes and comparing their results to the professors' evaluations. The tests will be refined until the two measures produce similar results.

The tests can then be used as criterion-based instruments to assess more creative techniques for assessing CT knowledge. Representative creative techniques were described in the previous section, and the team will develop rubrics for formalizing evaluations made with those. Results obtained using the rubrics will then be compared with those from the validated tests to assess the viability of both.

The timeline for these activities is presented in the Project Timeline on the next page.

Objective 3: Disseminate modules and rubrics via workshops for 4- and 2-year college faculty to build expertise and foster interdisciplinary collaboration between computing and music.

The structure of the workshops we propose was described in detail in the previous major section. We will host three-day focused workshops at our own institution. Our university owns excellent conference facilities (a former hotel), and there is easy driving access to our university from anywhere in New England. We are also well served by major airports in Boston, MA, and Manchester, NH, both of which are about 35 miles from our campus. Expenses to help defray costs for workshop attendees are built into our budget.

We will also offer introductory (half- or full-day) workshops in conjunction with conferences such as those sponsored by the Association for Computing Machinery (ACM) Special Interest Group in Computer Science Education (SIGCSE) [1] and the Association for Technology in Music Instruction (ATMI) [3]. These shorter workshops will introduce participants to our work as a means of enticing them to attend one of our three-day workshops.

As discussed earlier, we believe that having participants attend in interdisciplinary pairs is critical to the workshops' success. To accommodate people for whom it is impossible to come with a colleague from their own institution, we will endeavor to "match up" professors from complementary institutions who might establish an inter-institutional collaboration.

We plan to offer five three-day workshops over the three years of this proposal. One will be offered during the summer of the first year, while two will be offered in each succeeding year. The schedule for these workshops is presented in the Project Timeline that follows. We anticipate that we can accommodate about ten new pairs in each workshop, allowing us to reach 100 faculty members in up to 50 institutions over the three-year grant period.

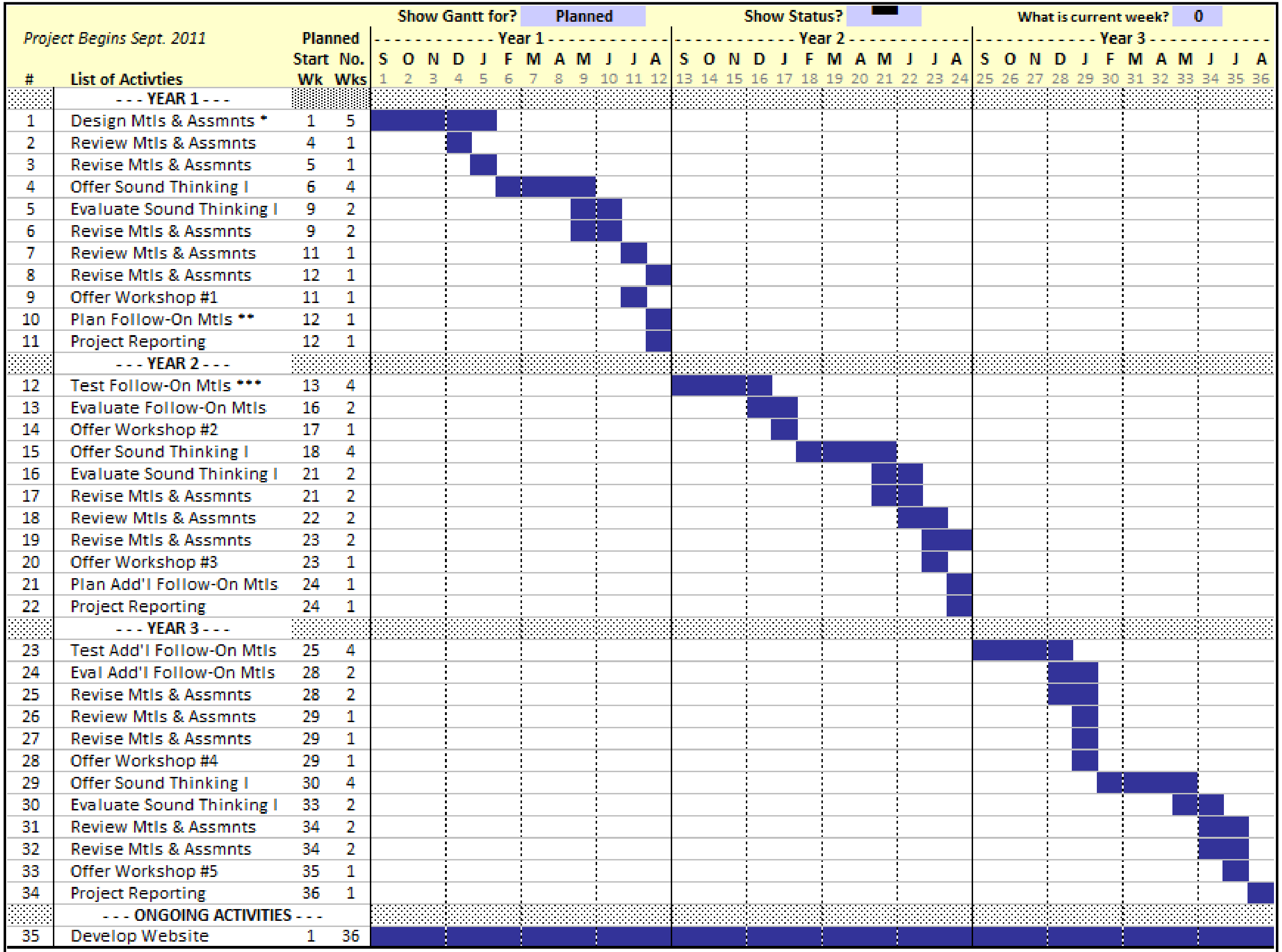
Project Timeline

We anticipate that this project will begin in September 2011. The Gantt chart [7] in Table 3 lays out the timeline for completing the activities discussed in previous sections.

How Will We Evaluate This Project?

Scott Lipscomb of the Univ. of Minnesota School of Music will serve as the external evaluator for this project. Scott is well qualified to fulfill this role because of his strong background in

Table 3. Project Timeline.



Show Gantt for? Planned

Show Status?

What is current week? 0

Project Begins Sept. 2011

Year 1: S O N D J F M A M J J A (weeks 1–12); Year 2: S O N D J F M A M J J A (weeks 13–24); Year 3: S O N D J F M A M J J A (weeks 25–36)

#	List of Activties	Planned Start Wk	No. Wks
	- - - YEAR 1 - - -		
1	Design Mtls & Assmnts *	1	5
2	Review Mtls & Assmnts	4	1
3	Revise Mtls & Assmnts	5	1
4	Offer Sound Thinking I	6	4
5	Evaluate Sound Thinking I	9	2
6	Revise Mtls & Assmnts	9	2
7	Review Mtls & Assmnts	11	1
8	Revise Mtls & Assmnts	12	1
9	Offer Workshop #1	11	1
10	Plan Follow-On Mtls **	12	1
11	Project Reporting	12	1
	- - - YEAR 2 - - -		
12	Test Follow-On Mtls ***	13	4
13	Evaluate Follow-On Mtls	16	2
14	Offer Workshop #2	17	1
15	Offer Sound Thinking I	18	4
16	Evaluate Sound Thinking I	21	2
17	Revise Mtls & Assmnts	21	2
18	Review Mtls & Assmnts	22	2
19	Revise Mtls & Assmnts	23	2
20	Offer Workshop #3	23	1
21	Plan Add'l Follow-On Mtls	24	1
22	Project Reporting	24	1
	- - - YEAR 3 - - -		
23	Test Add'l Follow-On Mtls	25	4
24	Eval Add'l Follow-On Mtls	28	2
25	Revise Mtls & Assmnts	28	2
26	Review Mtls & Assmnts	29	1
27	Revise Mtls & Assmnts	29	1
28	Offer Workshop #4	29	1
29	Offer Sound Thinking I	30	4
30	Evaluate Sound Thinking I	33	2
31	Review Mtls & Assmnts	34	2
32	Revise Mtls & Assmnts	34	2
33	Offer Workshop #5	35	1
34	Project Reporting	36	1
	- - - ONGOING ACTIVITIES - - -		
35	Develop Website	1	36

Notes

* "Mtls & Assmnts" refers to all course materials and assessments.

** "Follow-On Mtls" are course materials to be used in complementary, stand-alone courses that require Sound Thinking I as a prerequisite, or in music courses that are paired ("synchronized") with existing mid- to upper-level CS courses .

*** "Test Follow-On Mtls" means to use the new materials and assessments in stand-alone or synchronized courses.

both music and technology. He is the immediate past president of the Assoc. for Technology in Music Instruction (ATMI) [3] and is on the Advisory Board and serves as the Research Committee Chair for the Technology Institute for Music Educators (TI:ME) [38]. He is also the Editor of the *Journal of Technology in Music Learning*, a primary, peer reviewed, research-based journal. He has been involved in a number of projects that integrate music and technology and the evaluation of those programs [20], particularly in middle and secondary schools [21].

Scott will bring a fresh perspective to the evaluation of this project beyond that of the five professors at UMass Lowell who worked together on the *Performamatics* project. However, Sarah Kuhn of UMass Lowell, the evaluator for our *Performamatics* project, will collaborate with Scott to assist him and ensure continuity with our prior work. Sarah will also review forms and documents to ensure that we are in full compliance with all UMass Lowell Institutional Review Board regulations.

The evaluation will provide formative feedback to help the project be more successful as it moves forward and summative information about the extent to which the program has achieved its goal and component objectives. A number of tools will be used to achieve these ends.

Tracking Project Milestones

The project's objectives revolve around accomplishing specific milestones: the development of a suite of learning materials and assessments, the use of these materials in a defined number of classes, the development of a website for sharing materials developed by workshop participants as well as the project team, and the integration of that website with Ensemble, the NSF-sponsored distributed digital library for computing education [28]. The evaluator will work with the project team to determine the extent to which these milestones have been reached at the end of every semester. Particular difficulties or successes in doing so will be noted and appropriate adjustments will be made.

Course Surveys

Surveys will be distributed to students at the end of each course using the project materials, both at UMass Lowell and elsewhere. These surveys will be designed to elicit data on:

- how well — from a student perspective — the materials helped students acquire CT skills,
- what students thought might be improved about the courses and specific materials,
- whether student demographics have any relationship to the responses to the materials, and
- whether course participation changed student attitudes toward music and/or computing.

These surveys will be based on the survey developed for the *Performamatics* project, but modified to suit the specific materials used in each class. Workshop participants who use the materials during the dissemination phase will be encouraged to administer the same surveys and report their results to the project team.

Participant Follow-Up Surveys

Surveys administered immediately after a class cannot capture the long-term impact of that class on students' interest in computing and music creation, nor can looking at their academic records to see what classes they take in the future. This is because the types of changes this project encourages will not necessarily be reflected in immediate academic pursuits. We will therefore contact students still at UMass Lowell one, two, and three years after they take courses that use the project materials to have them complete surveys on their use of CT skills gained.

Faculty Interviews

Structured interviews will be held with the faculty members involved on a biannual basis, either in person or by phone, and with a sample of the workshop participants annually. Interviewees will be asked to comment on

- the impact of developing and using the project materials,
- the impact of interdisciplinary and cross-institutional collaboration, and
- what might be done differently to more positively impact student learning.

The interviews will be guided by a structured interview protocol, with additional questions asked as appropriate.

Tracking Website Usage

We will track usage of the project website by requiring users to register using a simple login and password routine that collects their names and affiliations and other relevant data before being allowed access to the downloadable materials. This will allow us to follow up with those users to collect data on the materials' use and effectiveness at their institutions.

Additional Evaluation Activities

In addition to the activities discussed above, Lipscomb will participate in meetings with the program team either face-to-face or via video conference. He will aid in annual reporting needs and provide data to be cited in articles or presentations by members of the project team.

What Are the Major Facets of Our Intellectual Merit and Broader Impacts?

Intellectual Merit

Our three objectives are complementary: develop, evaluate, and disseminate. Our approach to development is rooted in the interdisciplinarity of music and computing, such that both contribute to all the activities we use to teach. We design meaningful activities for the application of newly acquired knowledge so that all students, regardless of their major, deepen their understanding of CT through the engaging contexts of music and computing.

For example, a music transcription or composition activity may require the implementation of an array or list, techniques that CS majors learn early in their studies. The way in which the array is used in the musical domain, however, is most likely for a purpose that CS students will not have seen before or in a way that they have never considered, such as storing the note names as constants or MIDI values associated with a musical scale. The *application* of the technique in a new way not only expands students' understanding of the concept for which it is used, but also deepens their understanding of the technique itself.

Likewise, music majors or students who have significant musical training will gain new insights about music as well as computing when required to demonstrate their training in a computing environment rather than in a concert hall. Thus, even advanced students benefit from our interdisciplinary approach, because applying knowledge in new ways can deepen and enrich one's understanding just as much as acquiring that knowledge in the first place. We feel that the equalization of these seemingly disparate fields significantly enhances the intellectual merit of our work.

We also believe that our efforts to devise CT assessment vehicles contribute to the intellectual merit of this proposal. We have been unable to find any existing tools that address this issue, despite an extensive literature review, web searches, and e-mail discussions with other experts in the field. We understand the challenge of achieving this goal, but we feel that through the collective expertise of our project team and our workshop participants, we will make significant inroads into the problem.

Broader Impacts

We have been greatly encouraged by reactions to our prior work in *Performamatics* and its potential to assist students and researchers beyond our own university. We have experienced a high rate of acceptance of papers and other proposals submitted to conferences; have received numerous inquiries about our work and invitations to speak, serve on panels, and participate on committees; and, most recently, have been awarded a book contract by Oxford University Press. Statements in some of the letters of support included in the supplementary documents for this proposal also substantiate the widespread interest in interdisciplinary collaborations between music and computing.

The workshops that we now propose will be a primary vehicle for transforming all of this interest into action. Given the response we have seen to similar discipline-spanning workshops such as Guzdial's on Media Computation [11], Yanco's on Artbotics [46], and our own on Making Music with Scratch [17], we are confident that demand exists for the workshop we propose. In addition, we feel that the innovative structure we propose — in which participants attend in pairs from different disciplines and together develop materials they will use back at their own institutions — will ensure that a solid base is established for continuing involvement in the collaboration once the participants return home.

The most telling aspect of any program's impact is its sustainability over time. Our work at the intersection of music and computing continues to attract students and external interest, despite our original grant having now expired. *Sound Thinking* enrollments are higher this year than in previous years and include more students from outside music and CS. Our work has also contributed to the ACM SIGCSE Board's approval of a proposal by Bill Manaris of the College of Charleston, Bob Beck of Villanova, Jennifer Burg of Wake Forest, and Jesse Heines of UMass Lowell to establish a SIGCSE Music Committee, which will meet for the first time in March 2011 at the SIGCSE Technical Symposium in Dallas. Thus, we feel that we are already having an impact on CS education and believe that the research we now propose will significantly broaden that impact.

While these past successes demonstrate broader impacts than those solely at our own institution, we feel that the proposed research has even greater potential. Music is universal, and very much a part of youth culture. It provides cultural identity to minority groups not only in the US, but globally. We therefore believe that teaching music along with computing has huge promise as an approach to teaching CT, especially to underrepresented groups.

Finally, we see music as a *pathway* to computing. That is, we are proposing to go beyond offering just an introductory course by infusing music into more advanced courses in which students possess significant knowledge of underlying principles. Our approach will extend and deepen that knowledge through engaging, real-world applications. Such applications, distributed through Ensemble as well as our own website and introduced to faculty through comprehensive interdisciplinary workshops, will contribute to transforming undergraduate education in STEM.

What Experience and Capabilities Do the Participants Bring to This Project?

PI Jesse Heines, Prof. & Undergraduate Coordinator, Dept. of Computer Science, UMass Lowell

Jesse comes to CS with a background in educational media and technology. He has presented widely on the role of technology in CS education at ACM SIGCSE and other conferences, and he has had numerous papers published in conference proceedings. Jesse teaches courses in graphical user interface programming, web systems, and object-oriented programming, as well as co-teaches the interdisciplinary *Sound Thinking* course. He was PI on the NSF CPATH *Performamatics* award (CNS-0722161, $421,087).

Co-PI S. Alex Ruthmann, Asst. Prof., Music Education, Dept. of Music, UMass Lowell

Alex has published on technology-mediated music learning and teaching, children's musical and compositional creativity and creative processes, and fostering learner agency through music. He teaches undergraduate and graduate classes in music methods, qualitative and applied research methods, and creative processes and technology in music education. Alex develops new technologies for music learning and creativity with the Australasian Collaborative Research Center for Interaction Design.

Co-PI Gena Greher, Assoc. Prof. & Music Education Coord., Dept. of Music, UMass Lowell

Gena spent 20 years as a music producer/director in advertising before moving to UMass Lowell. She has published on the influence of integrating multimedia technology into the general music classroom, the middle school music curriculum, and the music teacher education curriculum. She teaches undergraduate and graduate classes in music methods, world music for the classroom, the socio-cultural impact on music teaching, and technology in music education.

Senior Person Fred Martin, Assoc. Prof. & Assoc. Chair, Dept. of Computer Science, UMass Lowell

Fred teaches courses in robotics, programming languages, software engineering, and artificial intelligence, and co-taught the *Performamatics* interdisciplinary *Tangible Interaction Design* course. In 2006, Fred received an NSF Faculty Early Career Development award (REC-0546513, $599,943). His present focus is science inquiry, using electronic sensors for data collection and the web for data-sharing and visualization.

Senior Person Sarah Kuhn, Prof., Dept. of Psychology, UMass Lowell

Sarah's commitment to innovation in learning, particularly blending technical education with the social sciences and arts, led her to create the UMass Lowell Laboratory for Interdisciplinary Design. She is a member of the Social Science Advisory Board of the NSF-funded National Center for Women & Information Technology and was a member of the National Research Council Committee on Workforce Needs in Information Technology. She was Co-PI of Project TechForce, an NSF-funded study of women and men working in the Massachusetts software and Internet industry.

Consultant Scott Lipscomb, Assoc. Prof., Music Education, Dept. of Music, Univ. of Minnesota

Scott teaches courses in music education, music cognition, and music technology. His research areas include facilitation of music learning through technology integration, interactive instructional media development, and multimedia cognition. His work has been published in numerous peer-reviewed journals and edited volumes, and he presents frequently at national and international conferences. Scott is Editor of the *Journal of Technology in Music Learning*, immediate past President of the *Association for Technology in Music Instruction*, Research Committee Chair for the *Technology Institute for Music Educators*, and Treasurer for the *Society for Music Perception and Cognition*.

REFERENCES CITED

[1] ACM SIGCSE (2009). *ACM Special Interest Group on Computer Science Education*. www.sigcse.org *accessed* Dec. 24, 2009.

[2] Allan, W., Coulter, B., Denner, J., Erickson, J., Lee, I., Malyn-Smith, J., & Martin, F. (2010). *Computational Thinking for Youth*. The ITEST Small Group on Computational Thinking White Paper Working Group. itestlrc.edc.org/resources/computational-thinking-youth-white-paper *accessed* Dec. 9, 2010.

[3] ATMI (2009). *About the Association for Technology in Music Instruction*. www.atmionline.org/index.php/about-atmi.html *accessed* Dec. 19, 2009.

[4] Brown, A.R., & Sorensen, A.C. (2009). *Interacting with generative music through live coding.* Contemporary Music Review **28**(1):17-29.

[5] Bukvic, I.I. (2009). *L2Ork*. l2ork.music.vt.edu *accessed* Jan. 5, 2010.

[6] Davey, B. (1983). *Think aloud: Modeling the cognitive processes of reading comprehension.* Journal of Reading **27**(1):44-47.

[7] Duggirala, P. (2010). *Gantt Charts – Project Management Using Excel*. chandoo.org/wp/2009/06/16/gantt-charts-project-management/ *accessed* Dec. 7, 2010.

[8] Encyclopædia Britannica (2010). *étude*. www.britannica.com/EBchecked/topic/194671/etude *accessed* Dec. 4, 2010.

[9] Ericsson, K.A., & Simon, H.A. (1998). *How to study thinking in everyday life: Contrasting think-aloud protocols with descriptions and explanations of thinking.* Mind, Culture, & Activity **5**(3):178-186.

[10] Greher, G.R., & Heines, J.M. (2009). *Sound Thinking: Conceptualizing the Art and Science of Digital Audio for an Interdisciplinary General Education Course*. Assoc. for Technology in Music Instruction. Portland, OR.

[11] Guzdial, M. (2010). *Project "Georgia Computes!" Summer 2010 Media Computation Workshops*. home.cc.gatech.edu/gacomputes *accessed* Dec. 17, 2010.

[12] Hansman, C.A. (2001). *Context-Based Adult Learning.* New Directions for Adult and Continuing Education **2001**(89):43-52.

[13] Heines, J.M., Goldman, K.J., Jeffers, J., Fox, E.A., & Beck, R. (2008). *Interdisciplinary approaches to revitalizing undergraduate computing education.* Jrnl. of Computing in Small Colleges **23**(5):68-72.

[14] Heines, J.M., Jeffers, J., & Kuhn, S. (2008). *Performamatics: Experiences with Connecting a Computer Science Course to a Design Arts Course.* Int'l. Jrnl. of Learning **15**(2):9-16.

[15] Heines, J.M., & Greher, G.R. (2009). *Course Website for 91.212/73.212 Sound Thinking, Spring 2009*. teaching.cs.uml.edu/~heines/91.212/91.212-2008-09s *accessed* Jan. 1, 2010.

[16] Heines, J.M., Greher, G.R., & Kuhn, S. (2009). *Music Performamatics: Interdisciplinary Interaction*. Proc. of the 40th ACM Tech. Symposium on CS Education, pp. 478-482. Chattanooga, TN: ACM.

[17] Heines, J.M., & Maloney, J. (2011). *Making Music with Scratch*. Proc. of the 40th ACM Tech. Symposium on CS Education. Dallas, TX: ACM.

[18] Higgins, H.J., & Wiest, L.R. (2006). *Individual interviews as insight into children's computational thinking.* Australian Primary Mathematics Classroom, Mar. 22, 2006.

[19] Hoy, J. (2010). *Song flowchart for The Beatles' "All You Need is Love".* Created for a course assignment in "Sound Thinking".

[20] Lipscomb, S.D., & Scripp, L. (2008). *Connecting Research that Reflects the Evolving Role of Music in Education.* Teaching Artist Journal **6**(2):157-168.

[21] Lipscomb, S.D., & Doering, A.H. (2010). *Technology in music and art instruction*. In M.D. Roblyer & A.H. Doering, eds. (2010). *Intergrating Educational Technology into Teaching, 5th ed.* Allyn & Bacon: Boston, MA.

[22] Markham, S.A., & King, K.N. (2010). *Using personal robots in CS1: experiences, outcomes, and attitudinal influences*. Proc. of the 15th Annual Conf. on Innovation & Technology in Computer Science Education, pp. 204-208. Bilkent, Ankara, Turkey: ACM.

[23] Martin, F., Greher, G.R., Heines, J.M., Jeffers, J., Kim, H.-J., Kuhn, S., Roehr, K., Selleck, N., Silka, L., & Yanco, H. (2009). *Joining Computing and the Arts at a Mid-Size University.* Jrnl. of Computing Sciences in Colleges **24**(6):87-94.

[24] Martin, F., & Roehr, K. (2009). *Cultivating Creativity in Tangible Interaction Design*. Creativity & Cognition Conf. San Francisco, CA: ACM.

[25] Martin, F., & Roehr, K. (2009). *Couse Website for 91.119/70.105 Tangible Interaction Design, Fall 2009*. tid09.wiki.uml.edu/ *accessed* Jan. 1, 2010.

[26] MIT Scratch Team (2009). *Scratch*. scratch.mit.edu *accessed* Dec. 21, 2009.

[27] National Research Council Committee for The Workshops on Computational Thinking (2010). *Report of a Workshop on the Scope and Nature of Computational Thinking*. Washington, DC: National Academies Press. www.nap.edu/catalog/12840.html *accessed* Dec. 17, 2010.

[28] National Science Foundation (2009). *Ensemble Distributed Digital Library for Computing Education*. www.computingportal.org *accessed* Jan. 1, 2010.

[29] Nielsen, J. (1992). *Evaluating the thinking-aloud technique for use by computer scientists*. In H.R. Hartson & D. Dix, eds. (1992). *Advances in Human-Computer Interaction, Vol. 3*. Ablex Publishing Corp. p. 69-82.

[30] Resnick, M. (2007). *All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten*. Proc. 6th ACM SIGCHI Conf. on Creativity & Cognition, pp. 1-6. Washington, DC, USA: ACM.

[31] Resnick, M., Maloney, J., Monroyhernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). *Scratch Programming for All.* Comm. of the ACM **52**(11):60-67.

[32] Ruthmann, S.A. (2007). *Strategies for supporting music learning through online collaborative technologies*. In J. Finney & P. Burnard, eds. (2007). *Music education with digital technology*. Continuum Press: London. p. 131-141.

[33] Ruthmann, S.A. (2009). *Computational Zeppelin*. scratch.mit.edu/projects/alexruthmann/736779 *accessed* Jan. 5, 2010.

[34] Ruthmann, S.A., & Heines, J.M. (2009). *Designing Music Composing Software with and for Middle School Students: A Collaborative Project among Senior Computer Science and Music Education Majors*. Assoc. for Technology in Music Instruction. Portland, OR.

[35] Ruthmann, S.A., & Heines, J.M. (2010). *Exploring Musical and Computational Thinking Through Musical Live Coding in Scratch*. Scratch@MIT. Cambridge, MA.

[36] Ruthmann, S.A., Heines, J.M., Greher, G.R., Laidler, P., & Saulters, C. (2010). *Teaching Computational Thinking through Musical Live Coding in Scratch*. Proc. of the 41st ACM Tech. Symposium on CS Education. Milwaukee, WI: ACM.

[37] Smith, D.E. (1997). *Repeats, Second Endings, and Codas*. www.scenicnewengland.net/guitar/notate/repeat.htm *accessed* Dec. 25, 2009.

[38] TI:ME (2010). *Technology Institute for Music Educators*. www.ti-me.org *accessed* Dec. 12, 2010.

[39] Trueman, D., Fiebrink, R., & Snyder, J. (2010). *PLOrk: The Princeton Laptop Orchestra*. plork.cs.princeton.edu *accessed* Nov. 18, 2010.

[40] Urban, J.E., Heines, J.M., Fox, E.A., & Taylor, H.G. (2009). *Panel on revitalized undergraduate computing education*. Proc. of the 40th ACM Tech. Symposium on CS Education, pp. 69-70. Chattanooga, TN: ACM.

[41] van Someren, M., Barnard, Y., & Sandberg, J. (1994). *The think aloud method: A practical guide to modelling cognitive processes*. London: Academic Press.

[42] Wang, G., & Cook, P.R. (2004). *On-the-fly programming: using code as an expressive musical instrument*. Proc. of the 2004 Conf. on New Interfaces for Musical Expression, pp. 138-143. Hamamatsu, Shizuoka, Japan: National Univ. of Singapore.

[43] Wang, G., Bryan, N., Oh, J., & Hamilton, R. (2010). *Stanford Laptop Orchestra (SLOrk)*. slork.stanford.edu *accessed* Nov. 18, 2010.

[44] Wing, J.M. (2006). *Computational Thinking.* Comm. of the ACM **49**(3):33-35.

[45] Wing, J.M. (2009). *Computational Thinking.* Jrnl. of Computing Sciences in Colleges **24**(6):6–7.

[46] Yanco, H. (2010). *Artbotics Summer 2010 Educators Workshop*. artbotics.cs.uml.edu/index.php?n=Programs.WorkshopSummer2010 *accessed* Dec. 17, 2010.

[47] Yanco, H.A., Kim, H.J., Martin, F.G., & Silka, L. (2007). *Artbotics: Combining Art and Robotics to Broaden Participation in Computing*. AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education. Stanford, CA.