Computing in the Arts: A Model Curriculum

Bill Manaris, Renée McCauley, Marian Mazzone, William Bares
College of Charleston
Computer Science and Arts Departments
Charleston, SC USA
{manarisb | mccauleyr | mazzonem | bareswh}@cofc.edu

ABSTRACT

In this paper, we describe Computing in the Arts (CITA), an innovative interdisciplinary major, which combines computer science with artistic theory and practice. CITA is relatively easy to implement across a variety of higher education institutions, because it repurposes existing courses and resources in computer science and the arts. CITA is relatively easy to implement across a variety of higher education institutions, because it repurposes existing courses and resources in computer science and the arts. At our institution, this involves three tracks (music, visual art, and theatre). CITA incorporates four synthesis courses, one per academic year, which help students interweave the two distinct curricular experiences into a cohesive whole in preparation for their capstone project. CITA has been fully implemented at the authors' institution. In the three years of its existence, it has increased participation in computer science by 23% mainly from an underserved population (students interested in the arts). Approximately 40% of these students are women. While these students are not necessarily strong in mathematics, they tend to be strong in design and creativity, thus enriching the pool of computing professionals with people able to envision new technologies and surprising innovations. Such people are fully versed in computer science so they can easily collaborate with more traditional computing professionals such as software engineers and scientific programmers. We discuss our successes, and provide pointers for others to establish similar degree programs, including potential challenges and lessons learned.

Categories and Subject Descriptors

K.3.2 [**Computing Milieux**]: Computers and Education - Computer and Information Science Education

General Terms

Human Factors

Keywords

Computing, art, music, theatre, game programming, creativity, design, degree, major, curriculum, education

SIGCSE'14, March 5–8, 2014, Atlanta, Georgia, USA.
Copyright © 2014 ACM 978-1-4503-2605-6/14/03…$15.00.
http://dx.doi.org/10.1145/2538862.2538942

1. INTRODUCTION

Computing in the Arts (CITA) is an innovative, interdisciplinary Bachelor of Arts degree program, which integrates computer science and information technology with art theory and practice.[1] The CITA curriculum incorporates tested pedagogy in Computer Science (CS) and the Arts. It is based on proven curricula and structured synthesis approaches [1-5, 13, 14], and previous curricular experiences by the authors [1, 6-10, 15]. The CITA program provides a liberal arts and sciences experience because the students acquire the knowledge and skills to combine creativity in the arts with the tools and conceptual modeling systems of computing.

The CITA major consists of 50 credit hours of required coursework in computer science, mathematics, an art area (music, visual art or theatre) and CITA synthesis courses. The major is structured around several synthesis experiences, at the freshman, sophomore, junior, and senior levels, each involving computing and one or more arts areas. The sequence of synthesis courses is designed to vertically integrate computational thinking in the arts throughout the undergraduate experience.

2. MOTIVATION

The goal was to create an innovative interdisciplinary degree that combined computing and arts, and allowed students to pursue coursework usually sequestered into separate schools or divisions. For several years, we had offered courses such as Computing in the Arts, Computer Music, and Game Programming. These courses attracted students from across campus and the courses always filled up. Students often asked if there was a follow-up course that they might take.

Some faculty expertise existed in the Computer Science (CSCI) Department, and faculty in both CSCI and the Arts disciplines were interested in collaborating. Our beliefs were that:

- Artists and musicians increasingly need to understand the conceptual framework and possibilities of computing, but most did not have the necessary computational thinking skills. Twenty-first century artists want to understand computational tools and techniques so that they may incorporate them comprehensively and effectively into the creative process to achieve their artistic vision.
- Computer scientists increasingly need to incorporate creativity, aesthetics and design into their work. Art enriches computational thinking with creative and aesthetic

[1] This effort has been supported in part by an NSF TUES grant (DUE #1044861).

expression, resulting in new techniques, innovative products, improved problem solving, and original inquiry.

At the time this degree was conceptualized (discussion began in 2008), the computing community was discussing new models for undergraduate education in computing. Through this degree we realized that we could also:

- Address the computational thinking needs of the educated individual in the 21st century (on our campus and beyond); and
- Broaden participation in computing by attracting students who were not attracted to the traditional computer science degree program.

2.1 Gauging Student Interest

Prior to moving forward with implementing the CITA major, we conducted a survey of students in first-year seminar classes at our institution in fall 2009. These classes spanned various disciplines including Biology, Philosophy, Physics, Political Science, Psychology, and Teacher Education. We also surveyed students in a few classes in Music, Art History and Computer Science (for non-majors).

The survey asked:

> *If you were in high school, how likely would Computing in the Arts have been your choice of college major?*
>
> *Scale: 1 (low) – 7 (high)*

Three hundred fifty-four (354) surveys were returned providing the following data:

Totals for each choice (1 low, 7 high)							**Total**
1	**2**	**3**	**4**	**5**	**6**	**7**	
93	70	52	52	51	24	12	354
26%	*20%*	*15%*	*15%*	*14%*	*7%*	*3%*	*100%*

We interpreted these results as supporting our plan to move forward with the new major. In particular, we decided to view choices 6 and 7 (above) as indicative of true student interest. In other words, 36 of 354 students (or 10%) would have considered CITA as their major. Assuming an incoming freshman population of 2,500 students (average at our institution), this would translate to 250 students who would seriously consider CITA as their major. Even more conservatively, focusing only on choice 7 (i.e., 12 of 354 or 3.39%), then 85 incoming students would consider CITA as their major.

2.2 CITA Advisory Committee

A committee of faculty from CSCI and Arts disciplines formed to develop the curriculum. The committee worked collaboratively to plan and implement the curriculum, modeling some of the integrative activity we hoped to encourage in our students. Faculty from each discipline determined what content from their discipline was essential. While the program is clearly interdisciplinary in scope, a decision was made to house the degree in the Computer Science Department in the School of Sciences and Mathematics. (The Arts departments are in the School of the Arts.) Three years later, the committee continues to meet, and collaborate across social media.

3. CURRICULUM

CITA incorporates existing computer science courses from an ABET-accredited B.S. degree program. It also includes 7 synthesis courses that are interspersed throughout the curriculum to integrate the different bodies of knowledge – computer science and the students' chosen art disciplines.

As it builds upon existing courses, CITA is an easy-to-implement curriculum model for infusing undergraduate computing education across the liberal arts and humanities.

3.1 Foundations

In developing CITA, we explored various curricular models for integrating computing and the arts. These included Cornell University's *Computing in the Arts* (CitA) concentration (offered by the CIS department) [3], UC San Diego *Interdisciplinary Computing and the Arts* major (offered by the Music and Visual Arts Departments) [13], and Yale University's *Computing and the Arts* curriculum (offered by the CSCI department) [14], among others. Of these, we selected Yale's curriculum (Yale model) as the most flexible curriculum that could be adapted and extended. Like the Yale model, CITA reuses available resources (e.g., faculty and courses) as much as possible. This minimizes implementation costs, making it easier to adopt. We modified the Yale model to fit the constraints of a mainstream undergraduate institution. Specifically, the CITA curriculum

- incorporates CSCI courses from an ABET-accredited B.S. degree program; and
- introduces several synthesis courses throughout the curriculum to help mainstream university students integrate the two separate bodies of knowledge (CSCI and an art discipline).

Implementation of the CITA model can be accomplished minimally with the addition of four new synthesis courses. These courses are discussed in detail in sections 3.3.2.1 and 3.3.2.2.

3.2 Learning Outcomes

The CITA program is designed to provide a liberal arts and sciences experience – students will acquire the knowledge and skills to combine creativity in the arts with the techniques and conceptual modeling systems of computer science and information technology.

CITA is designed to prepare graduates to:

- model processes, particularly those in arts applications;
- investigate, visualize, speculate, and invent using computing and computational thinking;
- synthesize innovative software applications and media combining music, images, sounds, lighting and stage design, and other digital artifacts; and
- explore the potential of computational thinking and its influence on society.

The characteristics of CITA graduates overlap with those expected of Computer Science graduates, thus many of the learning objectives in our existing courses for CSCI majors were applicable. Specifically, select learning outcomes/objectives, specified in CC2008 & now in CC2013, for Algorithms and Complexity, Computational Science, Graphics and Visualization, Programming Fundamentals (subsumed by Software Development Fundamentals in CC2013), Discrete Structures, Human-Computer Interaction, Information Management, Platform-Based Development were found to be relevant [1, 2].

Moreover, the learning outcomes specified for several of our existing courses, which also conform to learning objectives in CC2008 and CC2013 and the requirements for ABET-accreditation, were appropriate to CITA. This enabled us to reuse existing courses in Math and Computer Science.

Table 1. Credit hour requirements by discipline

Credit Hours Required	Discipline
10	Computing in the Arts (CITA)
16	Computer Science (CSCI)
6	Mathematics (MATH)
18	Arts concentration (ARTS/H, MUSC, or THTR)
50	**TOTAL**

3.3 Curriculum details

The 50 credit hour program is an interdisciplinary Bachelor of Arts degree program with a blend of existing courses offered by Computer Science, Mathematics, Music (MUSC), Art/Art History (ARTS/H), and Theatre (THTR). (See Table 1.) The major is structured around several synthesis experiences (CITA courses), at the freshman, sophomore, junior, and senior levels, each involving computing and one or more arts areas. (See Figure 1.) Majors select an Arts concentration area from Visual Art (ARTS/H), Music (MUSC) or Theatre (THTR).

3.3.1 Computer Science and Math Requirements

CITA majors complete 16 credit hours of CSCI and 6 credit hours of math courses. Table 2 lists the existing CSCI and math courses that were folded into the CITA curriculum. The CSCI courses taken by CITA majors are the same ones taken by CSCI and other majors offered through the CSCI department. No new math or CSCI courses need to be developed to implement the CITA degree program.

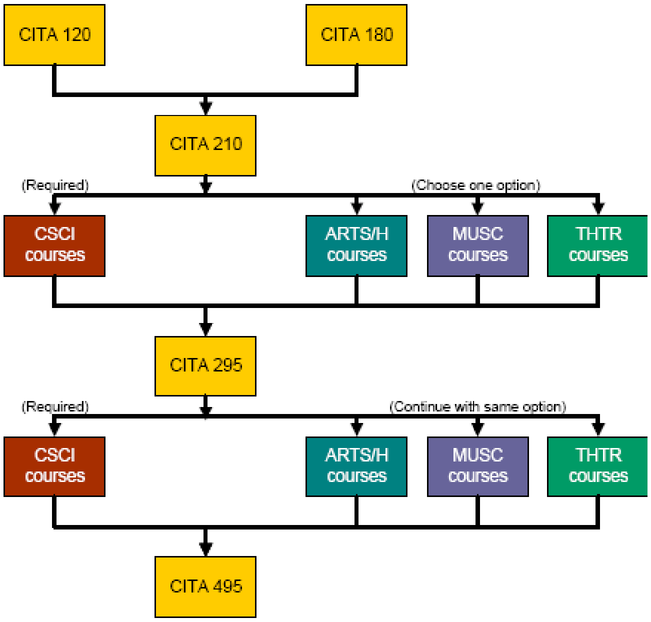

Figure 1: Curriculum overview by course type

3.3.2 CITA Requirements

The CITA synthesis courses are strategically placed throughout the curriculum to ensure a cohesive degree program for mainstream undergraduate students. As shown in Figure 1, students start with a sequence of two introductory synthesis courses. This ensures that they get an integrated foundation upon which to base their subsequent learning. A synthesis seminar is taken again in the middle of study, after students have taken several freshman and sophomore courses. Finally, a capstone experience is used to finalize the integration of the two bodies of knowledge (computing and their chosen art discipline).

Table 2. Computer science and math requirements

Computer Science

- Computer Programming I & lab (4 credits)
- Computer Programming II (3 credits)
- Three courses (9 credit hours) from:
  - Data structures and algorithms
  - Computer organization
  - Imperative programming languages
  - Declarative programming languages
  - User-interface development
  - Artificial intelligence
  - Computer graphics
  - Software architecture and design
  - Special topics courses, with permission

Mathematics

- One course from:
  - Pre-calculus
  - Business calculus
- Discrete mathematics

3.3.2.1 Introductory synthesis courses

Three CSCI courses that synthesize computing and the arts existed prior to the implementation of the CITA degree: CSCI 120, 180, 210 (descriptions below). These courses were originally created due to faculty interest and in an attempt to create CSCI courses that would attract majors, from across campus, into computing. Enrollments in these courses were healthy. When the CITA program was implemented, we cross-listed them as both CSCI and CITA courses: CITA 120, 180, 210. They are still cross-listed and continue to attract CS, CITA and other majors.

CITA 120: Building Virtual Worlds: This course synthesizes computational thinking with animation, storytelling, set design, movement through 3D space, lighting, etc. This course routinely includes guest lectures from Theatre faculty on topics such as costume design, lighting, choreography, and set design. (Students in the THTR concentration are encouraged to take this course.)

CITA 180: Computers, Music and Art: This course synthesizes computational thinking with creating music, sounds, images and other digital artifacts. The environments currently being used are MIT's Scratch, MIT's Processing, and jMusic with Jython, among others. This course routinely includes guest lectures from Art

History, Studio Art, and/or Music faculty on related topics. (Students in the MUSC and ARTS/H concentration are encouraged to take this course.)

CITA 210: Game Programming: This course introduces principles of game programming, including computer modeling, data visualization and animations, media transformations, and video game ethics. Students are exposed to several game engines, a scripting language, and develop at least one game. This course, in the guise of computer game development, exposes students to the synthesis of computational thinking with graphic design, sound design, animation, storyboards, programming, etc.

CITA majors choose one of CITA 120 & 180; all majors take CITA 210. Students completing the two-course sequence have the necessary computing background and motivation to place subsequent area concentration courses (Visual Art, Music, and Theatre) in a CITA context.

Since implementation of the CITA degree program, several new introductory synthesis courses have been developed (through funding from NSF). These new courses are described in Section 6. These courses were developed to make CITA into a national model curriculum; for this reason, they are listed separately. Some of these new courses are appropriate substitutes for CITA 120 or 180 in our own curriculum.

3.3.2.2 Upper-level synthesis courses

The following courses were developed to provide synthesis experiences at the junior and senior levels. These courses have been offered concurrently, once a year; a student cannot enroll in 295 at the same time as 495.

CITA 295: Seminar: Junior-level CITA majors participate in the seminar through readings, discussion, and individual library research to apply and expand their understanding of how to integrate computing and the arts. Students develop a prospectus on their future capstone project and identify a faculty advisor. This gives them a year to work on synthesis towards designing and implementing their capstone project. This course spans Bloom's Taxonomy levels 1-3 (Recall, Comprehension, Application).

CITA 495: Capstone: Senior-level CITA majors participate in the capstone through readings, discussion, preparation, and presentation of individual capstone projects under the guidance of the CITA advisor(s). This course spans Bloom's Taxonomy levels 4-6 (Analysis, Synthesis, Evaluation).

Successful completion of CITA 495 requires the development and presentation of a capstone project, and submission of a final report concerning the project. Projects must be approved by an arts advisor and the director of the CITA program. The project must synthesize arts and computing and must include software development (possibly incorporating open-source software, with permission of the instructor).

3.3.3 Concentration Requirements

There are three arts concentration areas in the curriculum: Art/Art History (ARTS/H in Figure 1), Music (MUSC), and Theatre (THTR).

The ARTS/H concentration requires 18 hours. This includes 6 hours of Studio Art and Art History courses, namely Drawing I, and New Media in Contemporary Art. It also includes 12 hours of electives from a list of major arts courses relevant to CITA, such as Themes in the History of Art, Introduction to Film Art, Drawing (II, III, and IV), Photography (I, II, III, and IV), Painting (I, II, III, and IV), Printmaking (I and II), Sculpture (I, II, III, and IV), The Cinema: History and Criticism, Studies in Modern, Contemporary & Film Arts, Contemporary Art, and Advanced Studio Practice. Prerequisite chains exist and must be satisfied.

The MUSC concentration requires 6 hours, namely Music Fundamentals, and Music Theory I. Additionally, it requires 12 hours of Music courses, from a list of major Music courses relevant to CITA, such as Music Theory (II, III, and IV), Composition (I and II), Music History (I and II), and Jazz Theory. Prerequisite chains exist and must be satisfied.

The THTR concentration requires 9 hours, namely Script Analysis, Acting I, and Playwriting I. It also includes 9 hours of electives from a list of major Theatre courses relevant to CITA, such as Costume (I and II), Stagecraft, Theatre History and Literature (I and II), Drafting and Rendering for Theatre, Creative Drama, Screenwriting, and Playwriting. Prerequisite chains exist and must be satisfied.

4. ASSESSMENT

After the three years of implementation, 56 students have declared CITA as their major. Of these, 8 have already graduated (by transferring earlier courses to satisfy the 4-year degree requirements). The remaining 48 students currently in CITA represent a 22% increase in students studying computing on our campus. This is significant because the majority of these students would not have chosen a computing major otherwise.

Of the CITA students currently enrolled, 40% (19 of 48) are female. In particular, the junior CITA honors synthesis class has 53% (7 of 13) female students; and the senior CITA capstone experience has 80% (4 of 5) female students. In other words, CITA, as implemented at the authors' institution, clearly attracts and retains female students. This should be compared to the recent 22% ratio of female students earning CS degrees nationally [11].

Enrollments have not yet stabilized, so it is unclear how these statistics will change in the future. We believe that, as our advertising efforts increase (we have done minimal marketing thus far, other than our web presence and one brochure), and word-of-mouth from our graduates spreads, our numbers will continue to grow, before they stabilize.

Comparing our current enrollments to our pre-implementation study, we find that of the estimated 85 incoming students that would consider the CITA major, about 22% of them actually matriculated. This translates roughly to 1% of our incoming freshman population. (Again, this is mostly a fresh, untapped population for computing, consisting of approximately 40% females.) Although rough (and not yet stabilized), we believe this first-three-years statistic may be useful to other schools considering implementing a CITA-like major.

5. LESSONS LEARNED

The CITA major and the CITA courses attract a broader population of students (in terms of gender) than our CSCI courses and majors have. Some students trying CITA courses, and being exposed to computer programming, have gone on to change their major to CSCI or CITA.

In the first three years of our CITA major, after various discussions among the CITA faculty constituents and through observations of student experiences and behaviors, we are considering the following improvements to our implementation of the CITA degree program.

Of most importance, we have concluded that all synthesis experiences (courses) should be three credit hour courses. Since inception, CITA 295 (the junior seminar) has been a 1 credit hour course. That gives the impression to some students that the course content and requirements, which includes deciding on a project topic and identifying faculty advisors, is not as important as their other classes. A change from 1 to 3 credit hours has already been approved by the faculty.

We want to recruit more arts faculty to work with CITA students as advisors. Several of our CS faculty are themselves artists, and can assist CITA majors as they creatively develop their project ideas. However, the combined talents from both CS and arts faculty acting as advisors will only strengthen a student's project. While being an art advisor does not require a major commitment of time, and in many cases will benefit the faculty member as much as the student, we have found that many art faculty prefer to remain within their traditional non-computing art constraints. A few notable exceptions (1 or 2 faculty per art area) have contributed to the forming of the CITA major, and are committed to advising and evaluating CITA students. However, we hope to gradually engage more (if not all) of the arts faculty, in one way or another, since there is much untapped talent in that pool of faculty. And as computing continues to spread through all arts disciplines, engaging with CITA is likely to become increasingly beneficial to arts faculty in terms of professional development.

The teaching of the capstone course can be taxing on the CS faculty involved, especially if some of the students are not as committed, or need a lot of support. Thanks to the web, in general, and YouTube and Vimeo, in particular, it is easy to have students find and present inspiring projects related to their interests. Requiring that they find research papers that have appeared in computing-related conferences helps identify projects that exist in the intersection of computing and the arts, and also have substance and promise to generate a capstone project. Many students have difficulty making the transition from YouTube (level 2 of Bloom's Taxonomy - understanding), to reading and discussing research papers (levels 3 and 4 of Bloom's Taxonomy - applying and analyzing), to coming up with a new, innovative idea that builds on earlier projects without replicating them (levels 4 and 5 of Bloom's Taxonomy - analyzing and synthesizing). Of course that is also a challenge in other situations where a thesis is expected (e.g., graduate school).

Evaluation of student capstone projects is another challenge, as the projects are often quite different in terms of content and the level of development can vary quite a bit. In particular, submitted work requires **analysis**, **evaluation**, and **creation** of CITA ideas, concepts, and materials into various deliverables, such as annotated bibliographies, project reports, and software (e.g., see [10, 12]). CITA graduates need to demonstrate their ability to model and apply process (i.e., to use programming skills) to achieve an artistic vision or project. We have found the following general guidelines to be useful in guiding the grading of projects:

- The grade of A is for work that involves high-quality achievement in all three Bloom areas.
- The grade of B is for work that involves high-quality achievement in at least two Bloom areas, and medium-level achievement in the other.
- The grade of C is for work that involves high-quality achievement in at least one Bloom area, and medium-level achievement in the others.
- The grade of F is for work that does not meet above criteria.

Advising/mentoring is time-consuming, but does not count as a class. So it adds to faculty workload without much recognition.

Finally, we have had some weaker CSCI majors switch to the CITA major, because they perceived it to be an easier degree (at least in terms of Math and CSCI requirements). However, these students may find creativity expectations of the capstone experience to be as equally challenging as the CSCI degree requirements. This is exacerbated by that fact that these students often switch into the curriculum late and attempt to compress all arts and synthesis courses into one or two years, which does not give them the time needed to synthesize the fields.

6. ADDITIONAL CITA COURSES

The CITA program allows us to explore possibilities for additional growth, and for strengthening ties between computing and the arts. For instance, through collaboration between CS and Arts faculty funded by NSF, we have developed the following new CITA synthesis courses:

CITA 101: #creativity (3 credit hours) This introductory course is an exploration of creativity and its applications for contemporary technology, business, art, and sciences. Students will gain understanding of the principles of creativity, participate in creativity exercises, and expand upon their creative skill-set. Small group projects, reviews of notable creative figures, examination of real world scenarios, and practical applications of creativity will foster the engagement and inquiry of creativity.

CITA 140: Drawing with Code (6 credit hours) This course seeks to provide the student with a historical, theoretical, aesthetic, conceptual, and technical introduction to the challenges presented by the relatively recent collision of art, culture, and computing power. Students will explore thinking algorithmically by deconstructing large ideas into programmable steps alongside lessons on how an artist thinks creatively through drawing.

CITA 160: Mobile Computing and Musical Theatre Fusion (3 credit hours) This course introduces musical theatre and computer science students to the fundamental principles of computer programming, playwriting, and song writing. Students will explore the intersection of technology and live theatre leading to ways that those ideas can be harnessed within the creation of a "mini-musical".

CITA 381: Computer Music on a Laptop: Composing, Performing, Interacting (3 credit hours) Principles of music composition and computer programming for developing interactive computer music environments. Team-based, project-driven exploration of Python programming, time-based structures, algorithmic processes, soundscapes, graphical user interfaces, musical language and style.

Syllabi and materials for all developed CITA courses are publicly available at http://www.cs.cofc.edu/~manaris/cita/

The following courses, which explore aspects of visual arts including animation or film, are in development:

Digital Sculpting and Modeling – This course starts with traditional modeling in clay or stone and moves to digital 3D modeling. Students create physical models then use a 3D scanner to digitize their work. They next learn to modify the shape and color properties of the scanned digital model. By the end of the course students will create digital models directly in the modeling software. Potentially, utilizing a 3D printer, they could print physical renditions of their artifacts.

Virtual Production – Previsualize and plan productions for television, film, and theatre through storyboards and modeling 3D digital stand-ins for scenery, props, and performers. Covers artistic conventions of visual storytelling and cinematography. Analysis and hands-on practice using emerging tools including performance capture, virtual

Compositing and Special Effects – Students learn how to use digital compositing software to blend live footage with digital backgrounds as well as to write their own algorithms to implement composites and special effects.

Once developed, course materials, including syllabi and exercises, for these courses will also be available through http://www.cs.cofc.edu/~manaris/cita/

7. CONCLUSION

We have presented Computing in the Arts (CITA), an innovative, interdisciplinary BA major, which introduces computing around a theme of creative expression. This major integrates computer science with art theory and practice. This CITA model is relatively easy to implement across mainstream institutions, as it reuses existing courses and resources in computer science and the arts. Only two new courses had to be developed to implement the curriculum. CITA incorporates four synthesis courses, one per academic year, which help students interweave the two distinct curricular experiences into a cohesive whole in preparation for their capstone project.

CITA attracts creative students who might otherwise have studied arts or the humanities. Finally, it attracts and retains about 40% female students compared to about 20% in traditional computing degrees [11].

8. ACKNOWLEDGMENTS

Development of the CITA curriculum model has been supported in part by NSF under a TUES (Transforming Undergraduate Education in Science, Technology, Engineering and Mathematics) grant, DUE #1044861). The authors acknowledge the significant support of the following on-campus collaborators: Chris Starr (Computer Science), Yiorgos Vassilandonakis (Music), Blake Stevens (Music), Charlie Calvert (Theatre), Jarod Charzewski (Studio Art), and Susan Kattwinkel (Theatre). Also, the following off-campus collaborators are acknowledged for various conversations on enhancing and disseminating the CITA model: Jennifer Burg (Wake Forest University), Susan Reiser (Univ. of North Carolina, Asheville), and Rebecca Bruce (Univ. of North Carolina, Asheville).

9. REFERENCES

[1] Computer Science Curriculum 2008: An Interim Revision of CS2001". ACM Interim Review Task Force. Retrieved Sep. 5, 2013 from http://www.acm.org/education/curricula/ComputerScience2008.pdf.

[2] Computer Science Curricula 2013, Ironman draft, http://ai.stanford.edu/users/sahami/CS2013/ironman-draft/cs2013-ironman-v1.0.pdf

[3] Cornell University, Computing in the Arts, Retrieved Sep. 5, 2013 from http://www.cis.cornell.edu/ComputingArts/

[4] Georgia Tech University College of Computing. 2013. "Welcome to 'Georgia Computes!' — Georgia Computes!" Retrieved Sep. 5, 2013 from http://gacomputes.cc.gatech.edu

[5] Guzdial, M. 2010. "Does Contextualized Computing Education Help?" *ACM Inroads*, 1 (4), pp. 4–6.

[6] Hanks, B., Murphy, L. Fitzgerald, S., McCauley, R., Thomas, L., and Zander, C. 2009. "CS1 Students Speak: Advice for Students by Students." *Proceedings of the 40th SIGCSE Technical Symposium on Computer Science Education*, Chattanooga, TN, pp. 19–23.

[7] Manaris, B. 2007. Invited Editorial, "Dropping CS Enrollments: Or The Emperor's New Clothes?" *ACM Inroads* (December), pp. 6–10.

[8] Manaris, B., Wainer, M., Kirkpatrick, A.E., Stalvey, R.H., Shannon, C., Levanthal, L., Barnes, J., Wright, J., Schafer, J.B., and Sanders, D. 2007. "Implementations of the CC'01 Human-Computer Interaction Guidelines Using Bloom's Taxonomy." *Computer Science Education Journal* 17 (1), pp. 21–57.

[9] Simon, B., McCauley, R., Morrison, B., Murphy, L. and Zander, C. 2009. "For Me Programming Is ..." *Proceedings of the 5th International Computing Education Workshop*, Berkley, CA, pp. 106–116.

[10] Starr, C.W., Manaris, B., and Stalvey, R.H. Bloom's Taxonomy Revisited: Specifying Assessable Learning Objectives in Computer Science, *Proceedings of the 39th ACM Technical Symposium on Computer Science Education* (SIGCSE '08), Portland, OR, pp. 261-265, Feb. 2008.

[11] Stross, R. What Has Driven Women Out of Computer Science? *The New York Times*, Nov. 15, 2008 (accessed online).

[12] Thompson, E., Luxton-Reilly, A., Whalley, J.L, Hu, M., and Robbins, P. *Bloom's taxonomy for CS Assessment, Proceedings of the 10th Conference on Australasian Computing Education*, vol. 78 (ACE '08), Wollongong, Australia, pp. 155-161.

[13] UC San Diego. ICAM: Interdisciplinary Computing and the Arts Major. Retrieved Sep. 5, 2013 from http://visarts.ucsd.edu/icam-interdisciplinary-computing-and-arts-major

[14] Yale University, Computing and the Arts. Retrieved Sep. 5, 2013 from http://www.yale.edu/yalecollege/publications/ycps/chapter_ii/computing_arts.html

[15] Zander, C., Thomas, L., Simon, B., Murphy, L., McCauley, R., Hanks, B., and Fitzgerald, S. 2009. "Learning Styles: Novices Decide." Proceedings of the 2009 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2009), Paris, France, pp. 223–227.