Designing Music Composing Software with and for Middle School Students: A Collaborative Project among Senior Computer Science and Music Education Majors

presented at the 2009 Conference of the

Association for Technology in Music Instruction (ATMI)

Portland, OR
October 22-25, 2009

Dr. S. Alex Ruthmann, Dept. of Music
University of Massachusetts Lowell
35 Wilder St., Suite 3, Lowell, MA 01854
Alex_Ruthmann@uml.edu

Dr. Jesse Heines, Dept. of Computer Science
University of Massachusetts Lowell
198 Riverside St., Lowell, MA 01854
Jesse_Heines@uml.edu

Abstract

What might be learned through a collaborative project bringing together middle school students, pre-service music educators and senior computer science majors in the development of new web-based music composing software for middle school students? This question framed a collaborative project undertaken by music education majors enrolled in General Music Methods II and senior computer science majors enrolled in GUI Programming II. Our demonstration will share examples of the software developed through the project as well as student compositions created by the middle school students who designed and tested the software. The challenges and surprises faced during the collaborative development process will be shared detailing our implications for future collaborations among our students. We will also present the rationale for our decision-making as professors. Finally, the lessons learned and insights gained by the middle school students and the music education and computer science majors will be shared and discussed.

Presentation Description

The music software used with students in K-12 schools is rarely designed with educational applications in mind. Audio production and composing tools such as Pro Tools, Sonar, Reason and Logic have extensive feature sets designed for professionals and are based on design metaphors influenced by studio production equipment and practices. While these software tools provide necessary flexibility and affordances in the professional setting, they can often constrain teaching and learning processes when adapted to school settings because of their complexity. Additionally, preservice music educators rarely have experiences in their preparation that help them understand music software design or to analyze the impact that technology design and

classroom implementation have on music teaching and learning. Similarly, undergraduate computer science majors often struggle in understanding the human factors of software design, particularly in aspect of graphical user interfaces and user interaction.

To help address these issues, a project was designed to bring together senior level undergraduate computer science majors taking a second-semester GUI programming course with senior level undergraduate music education majors taking a second-semester general music methods course. This project grew out of a pilot project developed within the scope of NSF/CPATH Award. That award was designed to integrate computer science courses with the arts in a program called Performamatics.

The students in the synchronized music education and computer science courses worked collaboratively on a project to design original music composition software for local middle school students. During the prior semester the music education majors had extended field experiences working with students at the local middle school. These field experiences continued throughout this project. Drawing on those prior and current experiences, the music education majors, along with the classroom teacher and students, collaboratively brainstormed ideas for new music software that would be tailored to the middle school students' interests and musical understanding. An additional social component to the project was developed where the middle school students would create, record and upload to a central server the musical sounds to be used as the compositional material in the various music composing programs developed by the computer science majors. Students in the GUI Programming class worked with these ideas and parameters in development of original software and an API for the final software projects. Throughout the development process, multiple versions of the software were tested by the middle school students and music education majors. This interactive feedback process led to further refinements of each software tool. Final projects were developed in Java as online web applications usable on any computer with a compatible web browser. This choice was made so that the programs could be easily tested and disseminated after the project.

As a result of this project, music education majors gained greater insights into the coding, design and capabilities of music software and how these aspects impact music pedagogical and curricular concerns. In particular, music education majors came to better understand how the constraints and affordances of music software impact students' composing processes and products. Through developing computer programs that would be used by middle school students with actual middle school students, computer science majors were forced to confront the human impacts of coding, design and interaction decisions, more accurately reflecting a real-world development environment. The middle school students and their teacher benefited through the use of new, custom-designed music software tailored to them.

Our demonstration will share examples of the original music software developed through the project as well as samples of student compositions created by the middle school students who were involved in the design and testing of the software. The challenges and surprises faced during the collaborative development process will be shared, detailing our implications for future collaborations among our students. We will also present the rationale for our decision-making as professors. Finally, the lessons learned and insights gained by the middle school students and the music education and computer science majors will be shared and discussed.

Acknowledgments

About the Authors

Alex Ruthmann is an Assistant Professor of Music Education at the University of Massachusetts Lowell where he teaches courses in music education, music technology and research. Dr. Ruthmann's recent publications include articles on classroom composing in the Music Educators Journal and Research Studies in Music Education as well as a recent book chapter on the use of Web 2.0 social networking tools in *Music Education with Digital Technology*, edited by John Finney and Pam Burnard published by Continuum Press. Dr. Ruthmann is a member of the editorial/advisory boards of the *International Journal of Music Education: Practice*, *British Journal of Music Education*, and the *Journal of Music, Technology and Education*. He currently serves as Chair-Elect of the Society for Research in Music Education (SRME) Creativity SRIG and is Managing Editor for the open-access *International Journal of Education & the Arts*. More information about his projects and research can be found at www.alexruthmann.com.

Jesse M. Heines is a Professor of Computer Science at the University of Massachusetts Lowell. He specializes in the implementation and evaluation of interactive, user-centered programs with rich graphical user interfaces (GUIs), particularly those employing Dynamic HTML, JavaScript, Java, and XML and XSL and their related technologies. Jesse has developed numerous computer-based instruction (CBI) programs and course websites as well as traditional human-computer interfaces. Prior to joining the UMass Lowell faculty, Jesse spent ten years with Digital Equipment Corporation, where he founded the Computer-Based Course Development Group and developed a large variety of CBI courseware. He holds a B.S. in Earth Sciences from the Massachusetts Institute of Technology, an M.S. in Science Education from the University of Maine, and an Ed.D. in Educational Media and Technology from Boston University. He has done post-doctoral work at The Open University in Great Britain, Brown University in Rhode Island, and the Massachusetts Institute of Technology.