Refactoring the DiVA: A Software Engineering Approach to Digital Music

Cameron Hassall, Nicolas d'Alessandro, Sidney Fels, Bob Pritchard
Media And Graphics Interdisciplinary Centre (MAGIC), University of British Columbia, Vancouver, Canada

Introduction

A Digital Ventriloquized Actor (DiVA) is a controllable speech-synthesis system. Sensors from multiple control devices capture the user's hand gestures and positions, which are then converted to phonemes (fundamental speech units). There are three main goals:

- Technical: a simple artificial neural network provides real-time input to a speech synth

- Scientific: researchers are interested in how DiVA users learn to speak with their hands

- Artistic: original musical pieces are written for and performed using the DiVA system

DiVA: Marguerite Witvoet   Designer: Hélène Day Fraser

Recent Work

The next version of the DiVA software was a translation of the original. Many components were unchanged. The user interface of the original Max/MSP version was preserved. Although object-oriented design principles were used, the system was still task-specific. It was designed and built for one purpose: to control and run the DiVA system. For the next major software revision, we desired a more modular system that could be extended to other speech- and music-related projects. The goal was a more flexible system that would still allow for the original research goals to be met.

Initial Work

The DiVA software consists of drivers for the input devices, mapping logic to convert hand gestures to phonemes, and a formant-based speech synthesizer. Initially, the DiVA software was written in Max/MSP, a visual programming language used for music-related projects. One of the disadvantages of this system was that it was difficult to debug. Users of the system experienced numerous crashes and system hangs. One of the crashes occurred during a performance. It was decided that the system would be rewritten in C++. The goal was a more robust system that would be easier to debug.

Conclusions

Translating the original DiVA software from Max/MSP to C++ resulted in a more robust system, suitable for performance. Object-oriented design ensured easier testing and debugging. The latest iteration of the DiVA software approaches a much more modular design. As well, we continue to improve the usability and expressiveness of the system by exploring new control methods and interface ideas.