Prototyping of Ubiquitous Music Ecosystems

Victor Lazzarini, Maynooth University, Maynooth, Ireland

Damián Keller, Amazon Center for Music Research (NAP), Federal University of Acre, Rio Branco, Brazil

Marcelo Soares Pimenta, Federal University of Rio Grande do Sul, Porto Alegre, Brazil

ABSTRACT

This paper focuses on the prototyping stage of the design cycle of ubiquitous music ecosystems. The authors present three case studies of prototype deployments for creative musical activities. The first case exemplifies a ubimus system for synchronous musical interaction using a hybrid Java-JavaScript development platform, mow3s-ecolab. The second case study makes use of the HTML5 Web Audio library to implement a loop-based sequencer. The third prototype—a sine-wave oscillator—provides an example of using the Chromium open-source sand-boxing technology Portable Native Client (PNaCl) platform for audio programming on the web. This new approach involved porting the Csound language and audio engine to the PNaCl web technology. The PNaCl Csound environment provides programming tools for ubiquitous audio applications that go beyond theWeb Audio framework. The limitations and advantages of the three approaches proposed—the hybrid Java/JavaScript environment, the HTML5 audio library and the PNaCl Csound infrastructure—are discussed in the context of rapid prototyping of ubimus ecosystems.

Keywords: *Computer Music, Csound, Music Programming Languages, Web Applications, Web Audio*

1. INTRODUCTION

Creativity-centered design of ubiquitous musical systems involves at least four developmental stages: defining strategies, planning, prototyping and assessment. This paper focuses on the third stage of the design cycle, prototyping. The first section shows related works in the field and the second places the activity of prototyping within the context of ubimus design. Then we present a case study focusing on the deployment of a ubimus system for synchronous musical interaction using a hybrid Java-JavaScript development platform based on browser technology. The second case involves the use of Web Audio in HTML5 to implement a loop-based sequencer. And the third case features a simple example of an HTML-controlled sine-wave oscillator using the PNaCl Csound programming environment. The final section provides a summary of the observations gathered during the design of these three prototypes and discusses the limitations and advantages of each approach.

DOI: 10.4018/JCIT.2015100105

2. RELATED WORK

In recent years, there has been some research (and commercial) work aiming to provide support for development of audio applications for mobile platforms like MobileSTK (Essl and Rohs, 2006), based on STK and released in 2006, with support for Symbian and Windows CE devices. This platform was also ported do iOS in 2010 (Bryan et al., 2010) and incorporated in a toolkit called MOMU. Also from Essl (Essl, 2010), we have Urmus, a LUA framework that is a multi-layered environment intended to support interface design, interaction design, interactive music performance and live patching on multi-touch mobile devices. Control (Roberts, 2011) is an application that allows users to define custom graphic interfaces for MIDI and OSC. The interfaces are defined using web standards like HTML, CSS and Javascript. Roberts also is one of the creators of Gibber (Roberts et al., 2013), a language for live-coding in the browser. Gibber also has a 2D drawing API and event handlers for touch, mouse, and keyboard events, enabling fast prototyping. Since Gibber is centralized on a server, users can create collaborative programming sessions and publish compositions and instruments.

3. DESIGNING UBIQUITOUS MUSIC SYSTEMS

Defining design strategies for ubiquitous music encompasses two areas of expertise: interaction and signal processing. The Ubiquitous Music Group (g-ubimus) has been investigating the musical applications of methods based on human-computer interaction and ubiquitous computing techniques. Metaphors for interaction provide abstractions that encapsulate solutions applicable to a variety of activities without making unnecessary technical assumptions (Pimenta et al., 2012). Thus, interaction metaphors materialize general ergonomic principles to fulfill the human and the technological demands of the activity (Keller et al., 2010; Pimenta et al., 2012). On a similar vein, recurring technological solutions can be grouped as interactions patterns (Flores et al., 2010). These patterns are particularly useful when developers face the task of finding suitable strategies to deal with specific interface implementation issues. So far, our group's research has unveiled four musical interaction patterns: natural interaction, event sequencing, process control and mixing (Flores et al., 2012). Each of these patterns tackles a specific interaction problem. Natural interaction deals with forms of musical interaction that are closely related to handling everyday objects. Event sequencing lets the user manipulate temporal information by freeing the musical events from their original time-line. Process control provides high-level abstractions of multiple parametric configurations, letting the user control complex processes by using simple actions. Mixing can be seen as the counterpart of event sequencing for synchronous interaction. Musical data—including control sequences and sound samples—is organized by user actions that occur in-time. Furthermore, technologically based musical environments also demand tailoring support for sound rendering. Signal processing techniques for creative musical activities have to be developed according to the characteristics of the tasks involved in the creative cycle, the computational resources provided by the support infrastructure and the profile of the target users. Ubiquitous musical activities may involve mobility, connectivity and coordination among heterogeneous devices with scarce computational resources. Thus, carefully chosen software design strategies are a prerequisite to tackle signal processing support in ubiquitous contexts (Lazzarini et al., 2012; Lazzarini et al., 2014).

Ubiquitous-music planning studies involve early assessment of target population expectations and identification of opportunities for creativity support. Through a ubimus planning study, Lima et al (2012) found sharply differing expectations on technological usage by musicians and