Project Proposal
Version 1.1

Sascha Zeisberger
Supervisor: Prof. Richard Foss
Computer Science Department, Rhodes University

1 April 2010

---

1 Principle Investigator

Sascha Franz Zeisberger
8 Cartright Avenue
Grahamstown 6120
082 372 0308
g07z3446@campus.ru.ac.za
Supervised by: Prof. Richard Foss

2 Project Title

Proposed Project Title:
Adapting and Simplifying UCMAN for the iPhone

3 Background

UCMAN is an application suite that allows audio connections to be established between audio devices over IEEE1394 networks. With the use of UCMAN, an audio engineer is able to share connections as well as control and monitor parameters between a variety of devices. By combining these properties with the use of firewire-based routers an audio engineer is able to construct large and relatively complex audio systems (such as those commonly found in stadiums and convention centers), and control them from a single interface running on the desktop environment.

UCMAN communicates with devices on a network using the XFN protocol, and interaction with this protocol has been simplified through the development of a class called UMANDLL[3]. UMANDLL handles all communications with

the network, including the discovery of devices, the joining of parameters among these devices and the streaming of audio signals.

The connecting of audio channels between devices is currently a four-stage process. First the two devices that are to be joined are selected. The multicores (audio channels) available for each device are then displayed where a transmitting and a receiving multicore is chosen for the source and the destination device respectively. After this is done the audio inputs are mapped to the multicores of the source device, and the multicores on the receiving device are mapped to one of its outputs.

Some of the underlying features UCMAN is built on rely on the Juce application framework, which is a set of classes that execute identically across a variety of platforms, such as that of Linux, Windows and Mac OS X. The network and GUI subsystems of UCMAN rely on this framework, as well as the Jucer component which allows for custom control elements to be created.

UCMAN also has an inbuilt skinning component called DeskItems that allows a manufacturer to define custom controls for the parameters of their device. The control parameters, such as volume faders and potentiometers of XFN-based devices can be defined with custom images with which a user can interact. This allows manufacturers to customize how a device is controlled and monitored.

4 Project Purpose and Objectives

In environments where an audio system may span large distances, such as convention centers and stadiums, the need for such a system to provide some form of mobility increases. Currently UCMAN is limited to the desktop environment providing a stationary point of presence, but it is deemable to port to the iPhone. When an application goes through a porting process of this nature various issues arise; is the platform able to handle the entire UCMAN feature set, will the screen size affect the usability level of the application suite and in what ways will the application need to be modified to effectively provide the same functionality as the original UCMAN application suite?

The questions posed can be summarised into a single problem to be addressed; in what way will the GUI and functionality of the mobile-based UCMAN application need to be modified to provide the maximum functionality on the mobile platform.

One proposed solution is that of a Plug-Oriented Patch Bay[4], which involves simplifying the four-stage parameter joining process into a single-stage process where the user simply selects the desired input and output plugs of the devices. This approach involves deriving a system that will scan and manage the

internal routings of devices and then automatically creates the connections between the desired devices. This system may prove to be useful in environments where audio engineering experience is limited, such as the home environments.

The main objectives of the project can be summarised as followed:

- Port the UCMAN application suite in its entirety to the iPhone.
- Modify the GUI for the UCMAN application suite for optimum usability on the smaller iPhone screen.
- Modify the application logic so as to simplify the functionality of UCMAN.

5 Approach

The development process for this project is divided into three main phases; Initial Studies, Investigation and Research, and Implementation.

The initial studies phase is involved in studying the base code for UCMAN, UMANDLL and gaining an understanding of the XFN protocol as well as the Juce framework. UML diagrams will be generated from the current desktop version of UCMAN in order to map the various sequences required when performing the application functions. Sources of information for this stage include the UCMAN and XFN source code, current documentation available and the development team for the UCMAN application suite.

The Investigation and Research phase is further divided into two phases; one dealing with design elements (GUI) and the other dealing with the application logic. Design elements and GUI design will need to be researched in order to develop a GUI that will provide both the functionality required of UCMAN and still be usable on the small-screened device. At the same time the application logic will need to be considered and has to compliment the final GUI design, allowing for similar functionality when compared to the desktop version of UCMAN. Performance will also need to be taken into account as the hardware present in the current generation iPhone may not provide the same performance as with the desktop version of UCMAN, and hence changes may need to be made in order to accommodate for this difference in hardware.

The implementation phase is involved in finally developing the application on the iPhone platform. What this entails is becoming accustomed to the XCode IDE and the Apple iPhone software development kit, modifying the existing iPhone UCMAN code to implement the modifications suggested in the investigation and research phase.

Additional phases beyond that of the three main phases include that of testing and extensions.

Testing will involve measuring the application performance using a variety of vectors. Performance in the sense of functionality will be quantified by measuring the difference in the number of functions available on both the desktop and the mobile version of UCMAN. Expanding on this test vector is stability and bug testing, attempting to remove any errors that arise through the use of the application. Application speed will be tested based on measuring the CPU, memory and network utilization. One important test vector unique to the iPhone and other touch screen devices is that of finger tracking accuracy. With the advent of touch screen technology came the issue of creating an interface that will accurately interpret commands and filter accidental clicks [1]. The accuracy of the touch screen technology employed in the iPhone will impact design decisions as to minimise the potential for accidental clicks.

The extensions phase will involve, time permitting, extending the project to incorporate any additional ideas and modifications that will further the UCMAN application on the mobile platform. This may involve performance modifications, feature extensions etc. and possibly larger extensions as mentioned in section 8 of this document.

6 Requirements

The following requirements are gathered from the iPhone SDK read-me file [2].

The hardware requirement include:

- Any Intel-based Apple System
- 1GB Ram
- (For the testing phase) Apple iPhone

The software requirements include:

- Apple Mac OS X Leopard (10.5.7)
- XCode IDE 3.1.3
- iPhone SDK for iPhone OS 3.1

7 Schedule

Project Deadlines:

Date	Activity
20 February	Project Website Initiated
26 February	Formal Written Proposal Completed and Accepted
2 March	Oral Presentation
21 May	Literature Review
20 July	Oral Presentation Update
17 August	Poster Presentation
25 October	Final Oral Presentation
1 November	Project Deadline
8 November	Project Website Completed

Proposed Schedule Milestones:

Date	Activity
10 February	Phase 1: Initial Studies *Study source code.* *Meet development team.* *Familiarise with XFN Stack.* *Develop UML Diagrams.*
30 March	Phase 2: Investigation, Research and Design *Redesign GUI to accommodate iPhone.* *Develop Patch-Bay logic (pseudo-code, UML diagrams).* *Benchmark UCMAN/iPhone performance.*
6 June	Phase 3: Coding and Implementation) *Start development in XCode Environment.* *Implement GUI and Patch-Bay System* *Implement required performance modifications.*
5 August	Phase 4: Testing, Write-Up and Revisions *Benchmark revised UCMAN version and modify as necessary.* *Begin compilation of study results and define conclusions.* *Modify project to incorporate any extensions (optional).*

8 Possible Research Sources

The following section provides a list of information sources that may be utilised throughout the phases of the project.

Apple. *iPhone SDK ReanMe*, 2009.

Chigwamba, N., Foss, R., Gurdan, R., Klinkradt, B. *An Intergrated Connection Management and Control Protocol for Audio Networks*, Universal Media Access Networks, October 2009.

Foss, R. *UNOS Creator - User Manual*, Universal Media Access Networks, 24 February 2010.

Foss, R. *UNOS Vision - User Manual*, Universal Media Access Networks, 18 February 2010.

Foss, R. *AES standard for audio applications of networks - Integrated Control, Monitoring, and Connection Management for digital audio and other media networks*, Rhodes University, 2010.

Foss, R. *AES informative document for the standard for audio applications of networks - Integrated Control, Monitoring, and Connection Management for digital audio and other media networks*, Rhodes University, 2010.

Foss, R. *XFN Application Programming Interface*, Universal Media Access Networks, 17 November 2010.

Otten, F. *UCMAN Notes*, Rhodes University, 18 March 2010.

9 Extensions

Possible extensions may include:

- Modifying the system so that it may track the iPhones position relative to the devices connected via the UCMAN system, and thus display the closest device's properties.
- Investigate a multi-touch interface for UCMAN so as to take advantage of the iPhone's multi-touch features.

References

[1] Diy touchscreen analysis, January 2010.

[2] Apple. *iPhone SDK ReadMe*, 2009.

[3] Chigwamba, N. Ucman information session. Rhodes University Computer Science Department.

[4] Foss, R. *Proposed Honours Projects*. Rhodes University Computer Science Department, 2010.