

CONTROLLING DISTRIBUTED AUDIO FROM A 3D GRAPHICAL USER INTERFACE (GUI)

Short paper submitted in partial fulfilment of the
requirements of the degree

BACHELOR OF SCIENCE (HONOURS)
COMPUTER SCIENCE

Rhodes University - Grahamstown



M. L. Hedges

2012

1. Introduction

As observed through the past few decades, many technologies have been adapted and transformed in order to be controlled by computers or computer based systems. This is convenient in a way that computers can be used to manipulate stored data, and use it in many ways to what it was used for previously. The distribution of audio is one of the technologies that has been digitized and is becoming increasingly popularly amongst venues that had distributed audio systems.

There are three main advantages that have been observed when using digital audio as opposed to the conventional audio systems, these are:

- **Cost:** For the bigger venues requiring larger distribution, the cabling required costs a large amount of money. In the case of digital audio networking, multiple channels of audio would be transmitted along a single Category 5 Ethernet cable. .
- **Weight:** Conventional cabling systems require a single cable per audio channel whereas digital audio is able to transport many channels of audio through a Category 5 Ethernet cable. The weight of a conventional system increases quickly as more channels are added, however there are no extra requirements for additional channels of digital audio. A 32 channel 'snake' cable with a length of 150 feet (46 meters) weighs approximately 32 kilograms which works out to an approximation of 0.7 kilograms per meter of cable. This is extremely heavy when compared to Category 5 Ethernet cable which weighs approximately ten kilograms per thousand feet (305 meters) which works out to 0.03 kilograms per meter of cable.
- **Availability:** Cable 'snakes' are manufactured by highly specialized audio companies. In contrast, category 5 Ethernet cable is readily available in almost any town or city in any country. It is a non specialized item and is used widely in businesses, educational institutions, households, and many more. At live concerts there is a high chance of cables being damaged as the venues are normally in open areas where the cables are vulnerable to various threats, so a broken cable could cause many problems.

2. Graphical Sound System Control

The field of graphical control is developing as rapidly as digital audio itself. Graphical sound control software allows a user to control the distributed audio

systems from a single centralized point. Controlling a distributed audio system in this manner is a lot consuming when it comes to resources than the conventional way of doing the control. The conventional way of controlling a system such as this would be to have specially trained technicians or sound engineers stationed around the venue each performing a highly specialized task. Digital audio control eliminates this as the devices around the venue can be controlled remotely from the single console. The purpose of control software is also to provide the user with a high level view of the system, where the computer takes care of any low level tasks - eliminating the need for any specialized training required as before.

There were four control programs that were researched throughout the course of this project. This involved inspecting their user interfaces and identifying any advantages/disadvantages they have.

2.1 Harman System Architect

System Architect is the software package developed by the Harman group to control HiQNet systems. HiQNet is the name of the protocol used in this application for controlling the audio devices. Sound control can be done once the user has set up the venue by following these four steps [11]:

1. Define the venue
2. Add devices by their location
3. Associating devices by their location
4. Routing the audio

System architect is aimed to provide the user with a high level of controlling the sound in a venue. This has the advantage that it provides an easy means of routing audio between rooms, but also has the disadvantage of the user not being able to select specific channels to connect.

2.2 BSS London Architect

London Architect is the software developed by BSS audio which uses the HiQNet protocol and allows for graphical sound installation [4]. It provides a standard set of audio devices which the user can add to the venue and create connections to other devices. This software provides a wide range of functions that allow things such as scheduling as well as automatic system adjustment to be performed [4].

London Architect is in many ways similar to System Architect. The major difference between the two is that System Architect allows connection between zones, whereas London Architect allows explicit connection between devices. The devices in London Architect shows the input and output jacks for the pre-defined list of devices, this allows the user to create specific connections whereas in System Architect that is taken care of by the software. These both have respective advantages and disadvantages and a trade off arises between the control the user has over devices, and the user friendliness of the system.

2.3 Biamp Systems' Tesira

Biamp Systems provides large scale audio solutions for venues with distributed audio. They also create their own products from the Digital Signal Processing (DSP) components to the software implementation [1]. Tesira is the successor for their previous software known as 'daVinci', which was their digital audio control software [1].

The main focus of Tesira was the scalability of solutions. It aimed to provide an easy means of controlling audio solutions that scaled from one room, to multiple buildings [3]. Tesira is able to control all the networked devices from a single terminal, however there can be more than one terminal used to control the devices.

The interface for Tesira is similar to London Architect with a more modern appearance. A feature that Tesira provides is a layered approach to the devices. This approach allows the user to isolate specific devices and only work with ones that are relevant to the current activity.

2.4 UNOS Creator

The core functionality of UNOS creator is the ability to create interfaces by the use of drag-and-drop instead of writing lines to manually code an interface [7]. UNOS creator provides the user with desk item capability to enable the user to specify images for various desk item controls. Each desk item has an associated parameter which is known as the "control parameter", this controls the value associated with the desk item [9].

The patching of devices can be done in UNOS vision which can be accessed from the main UNOS creator GUI. The subnet configurations are shown in a matrix view, which then allows the users to patch connections between the various devices.

Although UNOS creator allows a sufficiently low level method of controlling

devices, it does not offer any way of displaying a venue for device control. The devices are selected by their IP address and then they are able to be controlled by the user.

2.5 The need for 3D control

All the software mentioned above provide similar services to the end user which is controlling distributed audio from a centralized point. The applications that allowed venue based control only provided that from a 2 dimensional perspective. The purpose of this research was to investigate the use of a 3D modeling application (Google Sketchup) with programming capability in order to create software that could control devices from a 3D interface. The purpose of this would be to enable the user to create a highly detailed venue in which they could place devices and control them from there. This would allow a more user friendly approach where the user could move through the venue in real time.

3. Sound System Control Protocols

In order to control the devices a sound control protocol had to be implemented. The choice was narrowed down to three protocols which provided easy means of controlling distributed digital audio devices.

3.1 Harman HiQNet

HiQNet is the protocol developed by the Harman group, this is aimed to be a single solution for controlling all Harman networked devices [2]. This entire protocol is implemented my System Architect which is the primary program by Harman to control audio. This protocol has a tiered approach with the top level representing the actual device, each device must have at least one node manager contained in the device [5].

3.2 Crossfire Network (XFN)

The XFN protocol is an IP based peer-to-peer network layer [12] protocol [8] [10]. The main focus of the Devices on the network can be classified as "XFN Devices" if they host an implementation of the X170 stack in to allow the control. Each device has a 7 layered 'tree' structure which describes the device as well as the parameters inside the device. Parameters within the tree need to be addressed via the fixed seven level address [10].

3.3 AVDECC

The IEEE 1722.1 AVDECC (Audio/Video Device Enumeration, Connection, and Control) protocol specifies protocols for device management on an Ethernet Audio/Video Bridged (AVB) network as well as streaming. It is a Link Layer (Layer 2) protocol [8] [12] which was used for device control and audio streaming in this project. The reason this protocol was chosen is that it does not require specific implementation on each device, but works as a 'plug and play' protocol. One deficiency with this protocol at the time of implementation was that the sub-protocol for control had no version for windows (which was the OS the system was implemented in). This meant that devices could only be discovered and streams created, remote control of the devices could not be implemented yet.

4. Google Sketchup - A Tool for 3D design

Google Sketchup was the 3 dimensional modeling tool used for the the majority of the user interaction for this project. Sketchup is an open source modeling program which allows fast venue construction as well as Ruby programming capabilities [6]. Sketchup provides the user with a few simple, yet powerful tools for creating components in the 3D environment. These components can be grouped to form entities which can then have several operations performed on them manually, or through the Ruby API. Entities can either be created from scratch by the user, or downloaded off the Google 3D Warehouse. The 3D warehouse consists of an abundance of components that have been created and uploaded by other Sketchup users. Using these components speeds up the construction of a venue tremendously.

The programming platform provided by Sketchup is based on the language "Ruby". Ruby 'plugins' can be written by users and saved to the "Sketchup\Plugins" folder. These plugins are loaded on Sketchup startup and can then be used to perform operations within the program. Plugins have three broad functionalities which are performing actions such as adding tools, simplifying multi-step operations, and allowing the user to perform external operations.

5. Design of the 3D Graphical Sound System

The 3D sound system was designed using an iterative approach that included the use of Unifies Modeling Language (UML) diagrams. This involved analyzing

each stage of design in various steps, where each one was re-done several times to mitigate any flaws that were noted in the respective stage of design. The five stages that were undergone during the design stage are listed below.

1. **Requirement specification** - This provided a detailed description of the system that was required by the user.
2. **Use case diagrams** - These diagrams indicate the main functionality of the system and the external actors that interact with the system.
3. **Textual scenarios** - The textual scenarios were generated from the use case diagrams which show the events triggered by the users along with the responses by the system.
4. **Class diagram** - Class diagrams can be derived from the textual scenarios. This is done by identifying the nouns from the textual scenarios and deciding which ones were objects that could be described in the class diagram. The class diagram shows attributes, operations, and relationships between classes.
5. **Sequence diagrams** - Sequence diagrams describe the low level interactions between objects that are required to fulfill the system's goals

6. Implementation

6.1 The SkAv Protocol

The main focus on implementing the section was the interaction between the Visual C++ program and the Sketchup Ruby API. This was important as the sound control was based on the Visual C++ side and the interaction based on the Sketchup side. A protocol was developed named the SkAv (Sketchup/AVDECC) Protocol which defined a way of the two programs interacting. The SkAv messages were passed through TCP sockets which were connected to the address '127.0.0.1' (localhost) on port 30456 which allowed direct interaction between the two programs.

The SkAv messages are separated into commands that are three characters in length, separated by underscore characters. This simplified the parsing stage at each end as the message could be split upon the underscore character followed by the process of following the actions defined in the message. The source of the message is given as the first three letters, followed by any actions that the message needs to perform. Some messages end in either mac addresses or

error/success codes. The error and success codes used were taken from the Hypertext Transfer Protocol (HTTP), not all of them were used but rather ones that could relate to the interaction in the various cases.

Given below are two examples of SkAv messages. The example below shows one interaction between Sketchup and C++ which would take place during the device discovery stage. The C++ program sends the first message to Sketchup informing it that it has discovered a device and requests Sketchup to display it in the 3D environment. The second message shows the acknowledgment message from Sketchup given that the device has been enumerated successfully and is not a duplicate device.

CPP_ENA_ATT_00:22:97:ff:fe:00:40:9e

- **Source** - The source of this message is the C++ program, given by the first three characters "CPP"
- **Action** - The action of this message is to enumerate an audio device to the sketchup interface, given by the command "ENA". The second half of the action section of the message is "ATT" which is the abbreviation for Atterotech board, which is the type of device that should be displayed in the 3D environment
- **Mac Address** - The remainder of this message is the Media Access Control (MAC) address of the Atterotech Board. This is needed by Sketchup as it is a crucial detail in the construct used to store the list of audio devices.

SKP_ACK_ENA_ATT_00:22:97:ff:fe:00:40:9e

- **Source** - The source of this message is the Sketchup, given by the first three characters "SKP"
- **Action** - The action of this message is to acknowledge a previous message that has been sent, this is shown by the "ACK" action in the message. The rest of the message is identical to the action of the original message that invoked this procedure so this would ideally let the C++ program know which devices have been enumerated correctly and which haven't.

6.2 The XMOS Atterotech board

The Analog to Digital processing (ADP), and vice versa (DAP), was done using XMOS Atterotech boards. These are devices which have a separate set of audio

jacks for input/output, and an Ethernet port. These devices essentially fill the gap between digital and analog audio and are found at all the end points of the audio network. They each have a unique MAC address which is used when perform operations as the AVDECC protocol is a layer 2 protocol. These boards are controlled by XMOS chips, these consist of a mesh network of XCore processors. This is a scalable architecture as many XCores can be contained within a single chip [13].

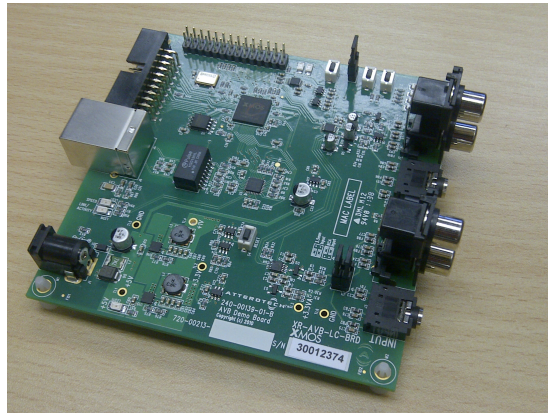


Figure 1: An XMOS Atterotech Board

7. Conclusion

The research for this project was successful as the system that was initially designed was implemented successfully. There is, however, some future work that is able to be done in order to extend the functionality of the system further. Some of the future work identified after the end result is given below.

- **Scaling the system** - The system worked for small audio distribution networks, however was unable to be tested for large networks. Scalability would be of utmost importance to test in both cases where the number of devices increases significantly (such as over 50), as well as the distance distribution of the devices.
- **Maximizing automation** - This goal was aimed at minimizing the interaction between the visual C++ interface and the user. Ideally all the interaction would be done through the Sketchup interface where the C++

application would run in the background and perform all the "behind the scenes" such as implementing the sound control protocol.

- **Implement the AEC** - The system current has no way of controlling the devices remotely as the AVCEDD Enumeration and Control Protocol has to be implemented to achieve this. This was not able to be done throughout the course of the project as there was no windows version of it at the time.

Bibliography

- [1] About Biamp. Online. Available from: <http://www.biamp.com/about>.
- [2] HiQnet Protocol and Harman System Architect. Online. Available from: http://www.ake.com/site/products/powerslave,id,948,pid,948,nodded,2,_language,EN.html.
- [3] Tesira. Online. Available from: <http://www.biamp.com/products/tesira/index.aspx>.
- [4] About London Architect. Online, 2010. Available from: http://www.bssaudio.com/LA_Features.php.
- [5] What is System Architect? Online, 2010. Available from: <http://hiqnet.harmanpro.com/about/>.
- [6] Sketchup Tools. Online, February 2011. Available from: <http://support.google.com/sketchup/bin/answer.py?hl=en&answer=73815>.
- [7] UNOS Creator. Online, 2012. Available from: <http://www.unosnet.com/unosnet/index.php/unos-core.html>.
- [8] DIBLEY, J. Audio networks, March-April 2012.
- [9] FOSS, R. *UNOS Creator User Manual*. UMAN, February 2011.
- [10] FOULKES, P. *An investigation into the control of audio streaming across networks having diverse quality of service mechanisms*. PhD thesis, Rhodes, September 2011.
- [11] HARMAN. *HIQNet System Architect 2 Workflow Overview*.
- [12] KUROSE, J., AND ROSS, K. *Computer Networking*, vol. 5. Pearson, 2010.
- [13] XMOS. XMOS Architecture. Online, 2012. Available from: <http://www.xmos.com/technology/architecture>.