

**A Remote Interactive Music
Keyboard Tuition System**

Mark Brian Newton

**A Remote Interactive
Music Keyboard Tuition System**

Thesis

Submitted in the fulfilment of the
requirements for the Degree of

MASTER OF SCIENCE

of

Rhodes University

by

Mark Brian Newton

April 2004

Abstract

A networked multimedia system to assist teaching music keyboard skills to a class is described. Teaching practical music lessons requires a large amount of interaction between the teacher and student and is thus teacher intensive. Although there is a range of computer software available for learning how to play the keyboard, these programs cannot replace the guidance of a music teacher. The possibility of combining the music applications with video conferencing technology for use in a keyboard class is discussed. An ideal system is described that incorporates the benefits of video conferencing and music applications for use in a classroom. A design of the ideal system is described and implemented. Certain design and implementation decisions are explained and the performance of the implementation examined. The system would enable a music teacher to effectively teach a music class keyboard skills.

Keywords: audio, video, MIDI, keyboard class, remote, interactive, multimedia

Acknowledgements

The completion of this thesis would not have been possible without the support, encouragement and patience of my supervisor, Richard Foss. I am indebted to him for his ongoing guidance throughout the period of this work.

I would like to thank my parents and, most importantly, my wife for their incredible patience, encouragement and help.

My thanks also go to Jenny Hallows who gave invaluable editorial advice.

Contents

1	Introduction.....	1
2	Current Status of Computer Technology for Teaching Electronic Keyboard Skills	4
2.1	Introduction	4
2.2	Background.....	5
2.2.1	MIDI Keyboards	6
2.2.2	Computers and MIDI	7
2.3	Available Educational Software for Developing Electronic Keyboard Skills.....	8
2.3.1	General Music Programs	9
2.3.2	Aural Programs	12
2.3.3	Music Theory Programs	13
2.3.4	Instrument Focused Programs	15
2.4	Summary.....	16
2.5	Concluding Remarks	17
3	Current Technology Applicable to Classroom Music Instruction	18
3.1	Limitations of One on One Music Lessons	18
3.2	Suggestions for Teaching a Class Keyboard Skills	19
3.2.1	Computer Software	20
3.2.2	Networked Computers.....	20
3.2.3	Specialised Education Systems	22
3.2.4	Video Conferencing	23
3.3	Summary.....	25
3.4	Concluding Remarks	26
4	An Interactive System for Teaching Keyboard Skills	28
4.1	Requirements for an Interactive System for Teaching Keyboard Skills.....	28
4.1.1	Teacher/Student Interaction	28
4.1.2	Audio, Video and MIDI	30
4.1.3	Music Teaching Software.....	31
4.1.4	Remote Desktop Sharing.....	32
4.1.5	Classroom Management.....	32
4.1.5.1	Class List	33
4.1.5.2	Teacher Assistance	35
4.2	Considerations for the Establishment of an Interactive System for Teaching Keyboard Skills.....	35

4.2.1	Ease of Use.....	35
4.2.1.1	Installation and Configuration	36
4.2.1.2	Class Use	36
4.2.2	Availability/Cost of Resources.....	36
4.3	Summary.....	37
4.4	Concluding Remarks	38
5	Design of a Remote Interactive Keyboard Teaching System	39
5.1	Design Approach	39
5.2	System Specification and Description	40
5.3	Use-Case Model	41
5.3.1	Finding the Actors.....	41
5.3.2	Use Cases	43
5.3.3	The Use Case Diagrams	45
5.4	Flow of Events.....	47
5.4.1	Student: Join the Class	47
5.4.2	Teacher: Add a Student.....	48
5.4.3	Student: Ask for Assistance	48
5.4.4	Teacher: Offer Assistance to a Student	48
5.4.5	Teacher: Transmit Multimedia Data	48
5.4.6	Student: Receive Multimedia Data.....	49
5.5	Classes and Objects	49
5.5.1	Finding the Objects	50
5.5.2	Filtering the List of Objects.....	50
5.5.3	Class Relationships	53
5.6	Sequence Diagrams	55
5.6.1	Student: Send Assistance Request.....	55
5.6.2	Teacher: Assist a Student	56
5.7	Summary.....	57
5.8	Concluding Remarks	58
6	Implementation of a Remote Interactive Keyboard Teaching System	59
6.1	Target Platform and Devices	59
6.2	Development Language and Tools	60
6.3	Graphical User Interface.....	61
6.4	Network Protocols	65
6.5	Multimedia Transmission	68
6.5.1	Audio.....	68
6.5.2	Video.....	69

6.5.3	MIDI.....	71
6.5.4	Application Sharing.....	74
6.6	Classroom Management	74
6.7	Performance.....	75
6.7.1	Computer and Network Performance.....	76
6.7.2	Delays.....	78
6.7.2.1	Audio	78
6.7.2.2	Video	80
6.7.2.3	MIDI.....	81
6.7.3	Synchronisation.....	82
6.8	Summary.....	83
6.9	Concluding Remarks	84
7	Conclusion.....	86
	Bibliography	88
	Appendices.....	91
A	Flow of Events	91
B	Sequence Diagrams	94

List of Figures

Figure 1: Screenshot of Cakewalk as an example of a sequencer.....	10
Figure 2: Screenshot of Band-In-a-Box	11
Figure 3: Screenshot of Auralia	12
Figure 4: Screenshot of Music Lessons I	14
Figure 5: WinSchool's class view.....	22
Figure 6: NetOpSchool teacher's view	23
Figure 7: One to one and multipoint models.....	24
Figure 8: Teacher and class interaction.....	29
Figure 9: One-to-one interaction.....	29
Figure 10: Teacher monitoring a student	30
Figure 11: View of the class lists	33
Figure 12: Assistance List.....	35
Figure 13: Teacher use case diagram.....	46
Figure 14: Student use case diagram.....	46
Figure 15: Relationships for the teacher side of the system.....	54
Figure 16: Relationships for the student part of the system.....	54
Figure 17: Sequence diagram showing student asking for assistance.....	56
Figure 18: Sequence diagram showing the teacher receiving the assistance request.....	56
Figure 19: Sequence diagram showing the teacher responding to the students request for assistance.....	57
Figure 20: Student programs graphical interface	62
Figure 21: Teacher programs graphical interface	63
Figure 22: Ethernet and UDP packet structures	66
Figure 23: Audio buffers.....	69
Figure 24: Source of MIDI messages on the teacher's side.....	71
Figure 25: Source of MIDI messages on the student's side.....	72
Figure 26: MIDIYoke connections	73
Figure 27: Performance Monitor graph.....	77
Figure 28: Sound delay	79
Figure 29: Audio delay	79
Figure 30: A video clip for the student's program.....	80
Figure 31: Student- Join the class	94
Figure 32: Student- Leave the class	94
Figure 33: Student- End of class	95

Figure 34: Student- Close multimedia communications.....	95
Figure 35: Student- Receives receiving request.....	96
Figure 36: Student- Receives multimedia data	96
Figure 37: Student- Transmit multimedia data	97
Figure 38: Teacher- Start the class.....	98
Figure 39: Teacher- Add a student.....	98
Figure 40: Teacher- Remove a student	99
Figure 41: Teacher- End of class	99
Figure 42: Teacher- Closes multimedia communications.....	100
Figure 43: Teacher- Start/stop multimedia transmission	101
Figure 44: Teacher- Transmit multimedia data.....	102
Figure 45: Teacher- Starts/stops multimedia receiving.....	103
Figure 46: Teacher- Receives multimedia data.....	104

List of Tables

Table 1: Actors for the design of the system.....	43
Table 2: List of use cases for the system.....	45
Table 3: List of possible objects	50
Table 4: Substituted nouns	51
Table 5: Boundary objects	52
Table 6: List of objects for the interactive keyboard teaching system.....	53
Table 7: UDP ports used for the multimedia data.....	67
Table 8: Attributes of telephone quality audio.....	68
Table 9: Student's control packet structure.....	74
Table 10: Teachers control packet structure	75
Table 11: Performance Monitor sections	76
Table 12: Bandwidth utilization.....	77
Table 13: CPU usage.....	78
Table 14: Results for the student's program	81
Table 15: Results for the teacher's program	81
Table 16: MIDI delay for the teacher.....	82
Table 17: MIDI delay for the student.....	82

1 Introduction

Creating expressive and entertaining sounds with instruments has been part of our history and culture for generations. The earliest trace of the use of musical instruments dates back to the Palaeolithic pipes and whistles, and Neolithic clay drums and shell trumpets [Britannica]. The use and scope of musical instruments has come a long way since then and nowadays, musicians and others interested in music making are harnessing the potential of computer technology to bring music making into the 21st century.

Learning how to play an instrument has not changed much over the centuries. People still learn how to play an instrument by being tutored or by watching others play, and then by taking the time to develop their instrumental skills. Private tutorship is the ideal situation in which to gain mastery over an instrument [Palmer, 1992]. However, the music pupil may end up paying dearly for the expertise and time of a qualified music teacher. There are many impracticalities of this tried and tested way of learning to play an instrument.

As new computer-related technology is becoming more widely available and affordable, computers are playing a larger role in being used to provide music education. People wanting to learn how to play an instrument can now make the most of available computer technology and related software. It is now possible to connect virtually any instrument to a computer, either directly or by using analogue to digital converters. For example, the electronic keyboard is a common instrument that is easy to connect to a computer.

The developments in computer technology have had a profound affect on music in general. Music studios are now equipped with computers for mixing and recording tracks and computers can also be used by the amateur as an aid to developing the skills necessary to become a proficient musician [Kirk, 1999]. Budding musicians can use sophisticated software to guide them when learning to play an instrument. Those choosing to learn how to play the electronic keyboard or a musical instrument have numerous software packages at their disposal.

An assessment of the status of software for teaching keyboard skills will show that despite the technological advancements, there are still many limitations to relying solely on software as a means of acquiring instrument skills. As sophisticated as the available software is for learning how to play the electronic keyboard, it does not offer the pupil the all-important human factor, that is, the need for human interaction, motivation and correction. Although

the software may contain an encyclopaedia of information and be able to recognize any note or chord played, it cannot replace the input of a professional and experienced music teacher. Unlike the teacher, the available software is not able to pick up bad technique, or suggest exercises that will keep a student encouraged and interested.

It is proposed in this thesis that a person can have access to and use software to learn how to play an instrument, combined with the benefits of one on one tuition. This thesis proposes and describes the development of an ideal system with the potential to be used in a classroom situation for electronic keyboard lessons. The system enables audio, video and MIDI to be transmitted across a network. A pupil or pupils wanting to learn how to play the electronic keyboard can have access to a music teacher in a class as though it were a one on one lesson. The design for this thesis combines aspects of video conferencing and MIDI for use in a distributed music lesson.

In Chapter 2, the music tuition software available will be discussed and how it can be used when learning how to play an instrument. Both the advantages and disadvantages of using the software will be examined.

Having shown both the advantages and disadvantages of the current state of instrument tuition, the possible combinations of the traditional and new technology will be considered in Chapter 3. The main question is if the strengths of the two approaches can be used to offer a better solution.

Based on the possible combinations, a description and design of a potentially 'ideal' solution will be presented in Chapter 4. It will incorporate the strengths of the tutor-based methods for learning how to play, along with the benefits that technology can offer.

In Chapter 5, a software design for the 'ideal' solution will be developed and discussed. Certain decisions will be explained when translating the design of an ideal solution using today's computer technology.

An implementation of the design will be described in Chapter 6. Various features of the system will be discussed in terms of the technology used and how the teacher and student would use the system. There will be a focus on the ease of use of the system since its primary role is to assist the teacher and student and not become a hindrance during lessons. The design and implementation of this system will be assessed in terms of how it meets the

original design requirements and description. The shortfalls of the system will be noted to allow for changes and increase its potential for commercial use.

Learning to play a musical instrument can be a rewarding and fulfilling experience. To continue in the tradition of the past by creating expressive and entertaining music using instruments to their fullest potential should not be limited by an inability to gain the best of human input and modern computer software. The 21st century musician can have the best of both worlds!

2 Current Status of Computer Technology for Teaching Electronic Keyboard Skills

2.1 Introduction

Today, new music students often choose the electronic keyboard as their preferred choice of an easy-to-learn instrument. For a beginner, keyboards can be easier to play than most instruments, and, as a result, music skills are learnt more quickly. Sheet music is also available for the keyboard in various styles of music, from classical compositions to modern chart hits. Electronic keyboards are widely available today and are versatile instruments that can produce a wide variety of sounds and mimic other popular instruments.

There are various ways of learning how to play the electronic keyboard, and the method chosen is determined by one's situation. Someone who wants to learn how to play the keyboard and learn about music has the option of attending a music class, being taught by a professional music teacher or teaching him or herself using books or other resources. However, it may not be possible to attend a music class owing to location, or because of the cost of the lessons, in which case, one can be self-taught.

A person seeking to learn how to play a keyboard¹ by independent means can use books, watch videos, listen to music tapes or watch experienced musicians perform. One could also simply start to explore and experiment with the sounds the instrument makes. Another possibility is to use computer software and take advantage of the connection between the electronic keyboard and computers. The method chosen to learn how to play the electronic keyboard would depend on what is available and what one wants to achieve as far as how proficient one wants to become on the keyboard.

The quality of software that is currently available for teaching one how to play the keyboard is related to the current state of computer technology. The development of electronic music and its connection to the computer have had a significant effect on the software available today.

¹ Electronic keyboard, keyboard, MIDI keyboard are often used interchangeably and can refer to the same thing, i.e. an electronic keyboard that has a MIDI interface.

The rest of this chapter will focus on available software that can be used to learn how to play the electronic keyboard. It is not a comprehensive review of the current state of music teaching software and related hardware, but rather, an overview of some programs available to music students wanting to learn how to play the keyboard. It covers programs aimed specifically at learning how to play the electronic keyboard, and some general music programs. The use of these programs is not limited to the home user; there are some that are better suited to a formal music class.

2.2 Background

Electricity has been used in musical instruments from as early as 1761, when J.B. Delaborde invented the electric harpsichord [Britannica], but it has only been since the 1940s and 1950s that electronics have been used extensively in musical instruments.

Early electronic keyboards were expensive and difficult to use. These instruments had to be programmed by the musician and constantly adjusted when used. The sounds they produced had a ‘synthetic’ nature and did not sound anything like the sounds produced by modern electronic keyboards. Electronic keyboards today are far simpler to use and they have sounds pre-programmed into them. Some of the sounds that are stored in electronic keyboards today are programmed to sound like other popular instruments, such as the piano and trumpet. Improvements to electronic keyboards are made continually so that they can produce a better quality of sound. Some of the more advanced electronic keyboards can emulate other instruments fairly accurately. As well as emulate other instruments, original and new sounds can be generated and used in compositions. Today electronic keyboards vary in functionality and cost to suit nearly any musical level and budget.

Keyboards can be very sophisticated pieces of equipment. Some have the facilities to record and play back musical notes that can then be used as a backing track. Expansion cards² with pre-recorded music on them can be connected to the keyboard. The more sophisticated keyboards have keys that are weighted to feel like a piano’s keys. The individual keys can be pressure sensitive so that the sound produced can vary as the pressure on them is changed.

² 1.44” disks, memory sticks, flash memory, compact flash, ...

2.2.1 MIDI Keyboards

A very important development for electronic instruments of today was the introduction of the Musical Instrument Digital Interface³ (MIDI) specification [Rothstein, 1992]. It consists of a simple hardware interface and a transmission protocol that enables electronic keyboards and other electronic instruments to exchange musical information. MIDI was developed to enable musicians to layer the sounds produced from different electronic keyboards in their performances. The layering of sounds was possible in a studio where tracks could be recorded separately and later combined, but this was not possible in a live performance. The musician was limited to playing one or maybe two keyboards at a time during a live performance.

What was needed was for one keyboard to be played and more than one keyboard to produce sounds at the same time. This was first demonstrated in 1983 at the first National Association of Music Merchants show [NAMM]. The MIDI specification was demonstrated by using two keyboards from two manufacturers to produce sounds simultaneously by playing either one of the keyboards.

The most common musical instrument with a MIDI interface is the electronic keyboard. The various functions of the keyboard are mapped into MIDI messages. These include program change messages for controlling the instrument selection, and note on and note off messages for representing different notes played on the keyboard.

The MIDI specification has some limitations that are becoming more restrictive as musicians' demands and expectations grow. There is a limit to the number of MIDI devices that can be connected to a single MIDI port, and the speed of the connection can be limiting, particularly for more complicated pieces of music [Rothstein, 1992]. However, these limitations would not be noticeable to the beginner who is new to learning how to play the electronic keyboard.

The MIDI specification opened up endless applications for electronic instruments and most profoundly, made the connection between music and the computer possible.

³ The acronym MIDI is now commonly used as a word; instead of saying an electronic keyboard with a musical instrument digital interface, one says a MIDI keyboard. Likewise, a computer has a MIDI available but one says a computer has a MIDI interface.).

2.2.2 Computers and MIDI

When the MIDI specification was developed, Roland Corporation introduced the Musical Processor Unit model 401 (MPU-401) [Roland]. The MPU-401 card allowed the popular IBM PC to communicate with other MIDI instruments. The computer was introduced as a controlling instrument, which helped drive MIDI towards becoming widely accepted. The development of the MPU-401 and similar interfaces for other computers allowed a standard and straightforward way to connect an electronic keyboard to a computer. The computer could now become a musical instrument that a musician could use.

Sound cards in computers do not typically have MIDI ports directly on the card, instead, they use a separate connector that plugs into the games port (joystick port, 15-pin, D-shaped connector) to provide the MIDI ports [Mueller, 1998].

Connecting a keyboard and setting up MIDI does require some technical ability, but is not beyond the capability of most home computer users. With the correct software, a computer can be used to record a sequence of MIDI events and play them back through a sound card.

Since the computer can play MIDI notes through the sound card, inexpensive MIDI controller keyboards can be used to generate MIDI messages without producing any sounds themselves. These keyboards need do nothing more than send MIDI messages in response to keys pressed on the keyboard.

It is relatively inexpensive to set up a PC with a MIDI keyboard and related software. As a user's skill develops and they need more from their PC, it can be improved incrementally without too much expense by adding a more advanced sound card, or more MIDI instruments.

Following on this capability of using the computer to record and play back MIDI notes, a wide variety of computer-based music programs have been developed and are being used. There are programs available that are used by professional musicians to compose and record their work, as well as programs that teach users how to play music.

2.3 Available Educational Software for Developing Electronic Keyboard Skills

Learning to play the electronic keyboard does include learning about certain fundamentals of music theory. The beginner musician needs to grasp elementary concepts of music theory. The need to acquire an understanding of music theory may not be explicit, but it is implied when learning how to play the electronic keyboard [Taylor, 1990]. Being able to recognise music notes and knowing the relationship between these notes and the music keyboard is important. An understanding of other elements such as rhythm, timing and harmony are also needed to be able to play the keyboard. These elements of music theory are sometimes not addressed as a topic, but are learnt by a beginner when practicing and learning how to play.

The types of music programs available for the computer vary from multimedia reference works, to specialised programs for composing and arranging musical scores that are used by professional musicians. There is a variety of software available that can be used to learn how to play the keyboard. These programs are aimed at different age groups and skill levels and the lessons are designed for the different types of users. Some programs are light-hearted, while other programs are more serious.

A beginner can use programs such as sequencers to explore music composition, mixing and sequencing. Sequencers are used for recording, editing, and laying out the different tracks for MIDI songs. There are far too many programs available to mention them all, but some of the more popular ones will be considered in this thesis.

Software that can be used to teach music and keyboard skills falls under the topic of Computer Aided Instruction (CAI). This topic will not be looked at in detail in this thesis. What is of interest is what is available, how it is used, and how it can be used by someone wanting to learn how to play the electronic keyboard.

The following section of this thesis discusses music software according to function. There are no clear distinctions between different types of music programs that can be used for learning how to play the keyboard, as some programs are not related specifically to learning how to play the keyboard, but can easily be used by a new student to explore music and develop musical skills. These programs range from applications used to transcribe musical scores and print them out, to sequencers that can be used to record and mix multiple sound tracks. There are also programs that can be easily used to generate different styles of music.

The various types of programs have been grouped into four categories and are discussed in more detail in the following sections – General Music Programs, Aural Programs, Music Theory Programs and Instrument Focused Programs. A music student at home could use a selection of the programs to learn how to play the keyboard, or music teachers could use them as a teaching aid.

2.3.1 General Music Programs

There is a wide range of programs that have been included in this group. They vary from reference programs to those that explain how to set up MIDI on a computer. These programs may not be the first choice of someone wanting to learn how to play the keyboard, but they can contribute to the learning experience by providing additional interesting and useful information, and encourage the user to experiment with music.

Reference programs are text-based and include multimedia⁴ samples as illustrations, for example, *Discovering Music*, *Multimedia History of Music* and *Multimedia History of Instruments* [Voyetra]. These programs are part of Voyetra Value CD Series, although discontinued they can still be found in some shops. Reference programs offer a resource of information about music and instruments. They can also serve as a good introduction to the world of music.

Along a similar theme are live-to-MIDI recordings, such as *The Pianist Performance Series* [PGMusic], which looks in depth at different styles of music played on a piano. Each program contains biographies of musicians, an on screen piano display and music quizzes. The styles covered include Blues, Jazz, Latin and New Age. When the songs are played, they can be slowed down so that each individual note played on the keyboard is displayed on the screen. Other MIDI programs, such as sequencers, can use the MIDI files that are available on these compact discs.

A very useful type of program for anyone using a MIDI keyboard connected to a computer is the sequencer. Anyone wanting to learn how to play the electronic keyboard would benefit from having a good MIDI sequencer available. A beginner could use the sequencer to record

⁴ Multimedia – refers to text, font, colour, pictures, sounds and video.

their playing, and then play it back to hear what it sounds like. They can then edit the recorded music and start experimenting with the composition. The student can also see the individual notes as they play a MIDI file.

The more popular sequencers include *Cakewalk* [Twelve Tone Systems], *Powertracks* [PGMusic], *VST* [Steinberg] and *Orchestrator* [Voyetra]. A screenshot of a demonstration version of *Cakewalk Professional* is shown in figure 1. All these programs offer multi-track recording, editing, and play back of MIDI files. There are differences in the functionality and unique limitations of each program, but these are not usually relevant to a beginner. Music can be entered into these programs in various ways including Piano Roll, MIDI event editor, music notation, loaded from a MIDI file, or recorded from a MIDI keyboard.

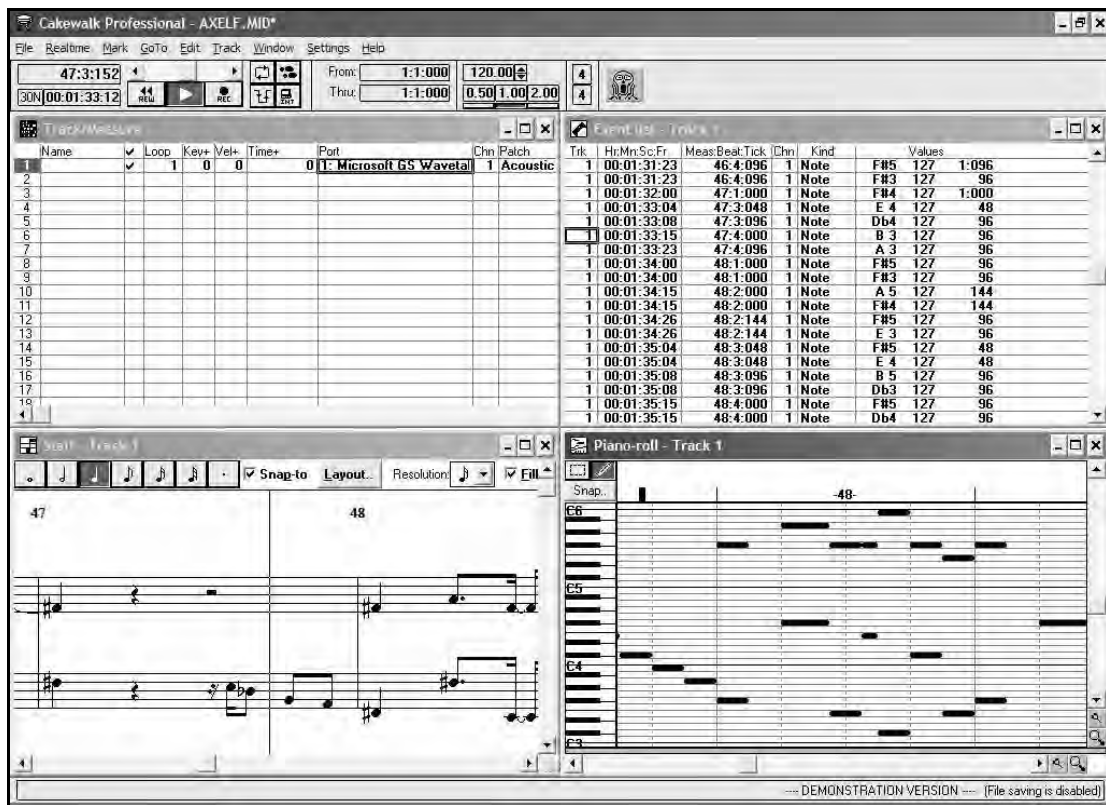


Figure 1: Screenshot of Cakewalk as an example of a sequencer

Another type of program that would be useful for a new keyboard player is a notation program. As was mentioned earlier, being able to read and write music is important when becoming proficient at playing the keyboard. The more popular notation programs can read and write MIDI files, they can load a MIDI file and display it in musical notation, as well as save a composition as a MIDI file. The MIDI files can be used in other programs such as

sequencers. It may not be necessary for a beginner to have a separate program for notation, since most sequencers have the option of viewing and editing music in notation view.

A popular and highly regarded music notation program is *Sibelius 2* [Sibelius]. Others include *Finale* [MakeMusic], *Allegro* [MakeMusic], *PrintMusic* [MakeMusic], and *MusicWrite* [Voyetra]. For a beginner, *PrintMusic* or *Allegro* would be more than sufficient. The others tend to be more complex and offer some advanced features for which a beginner would not have any use. It is possible to play a piece of music on the keyboard and record it using the above mentioned notation programs. The music can then be edited and saved or printed out.

Band-in-a-Box [PGMusic] could be considered in a group of its own. Out of all the programs discussed, it is arguably the most enjoyable program for a beginner. Figure 2 below shows a screen shot of *Band-In-a-Box*. By simply typing in the chords and choosing a style, *Band-in-a-Box* generates musical backing comprising of different instruments. The music can then be played back, saved, or edited, in piano roll or music notation. This program allows a beginner to explore different styles of music and chords easily.

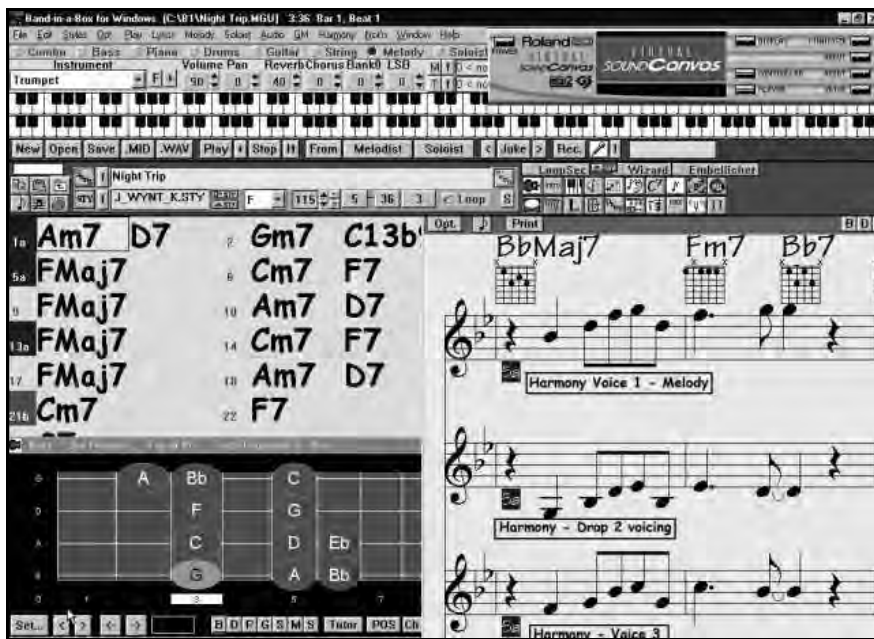


Figure 2: Screenshot of Band-In-a-Box

Other programs that may be useful to beginners are *SoundCheck* [Voyetra] and *Computer Music Starter Kit* [Voyetra]. These programs are similar in that they are aimed at how to connect a MIDI keyboard to a computer. *SoundCheck* is a program that can be used for

troubleshooting audio and MIDI set-ups. It also has a tutorial on how to connect a MIDI keyboard to the joystick port on the computer. It includes a video with step-by-step instructions for connecting a MIDI keyboard to a computer. *Computer Music Starter Kit* includes the cables and instructions for connecting a MIDI keyboard to a computer.

2.3.2 Aural Programs

A further broad group of programs is aimed at developing aural and vocal skills. Again, not necessarily directly related to learning how to play the keyboard, but very useful, particularly for developing an awareness of the sounds of different notes. These programs can be used to supplement programs aimed specifically at keyboard skills. There are also numerous commercial and shareware programs available for aural training. The more popular and better programs include *Auralia* [Rising Software], *EarMaster* [MidiTec], *Music Lab Harmony* and *Music Lab Melody* [MusicWare].

Auralia, shown in figure 3, is an ear-training course that can be used with a microphone or MIDI keyboard. It has many drill-based exercises and the results of the exercises can be saved to monitor the student's progress. It covers topics such as intervals, scales, rhythm, melody and pitch.



Figure 3: Screenshot of Auralia

In *EarMaster*, a user can choose which exercises to practice, or follow the set course. The program offers three instrument views for giving answers, namely: piano view, guitar tablature or a note sheet. As in *Auralia*, all the results from the exercises can be saved and reviewed.

The Music Lab Series includes *Music Lab Melody* and *Music Lab Harmony*. These two programs cover how to hear, read and write music. They offer many exercises with immediate feedback, and results can be stored for each user of the program. *Music Lab Melody* is aimed at people who are learning how to read and write music and who need to recognise and notate musical notes. *Music Lab Harmony* focuses on more advanced aspects of music, and therefore is not particularly suitable for a beginner. It covers harmony, recognition of music intervals and a progressive study of harmony, including keyboard harmony skills.

2.3.3 Music Theory Programs

The next broad category of programs teaches music theory. These programs may not necessarily be an obvious choice for those wanting to learn how to play the keyboard, however the keyboard is used in these programs for demonstrations and exercises. The programs generally aim to develop fundamental music skills, including sight-reading, note recognition, a sense of rhythm and melody, chord recognition and scales.

Music Lessons I, *Music Lessons II* [MiBAC] and *Musition* [Rising Software] are the most popular music theory programs. They cover a wide range of topics and have numerous drill exercises that offer immediate feedback.

Music Lessons I, shown in figure 4, is aimed at students of any level and is most suited for the beginner, as it does not require any musical background. The program supports MIDI and it can be used when learning how to play the keyboard. A comprehensive online help system is available that explains every term used in the program. *Music Lessons I* covers note recognition, sight-reading and other aspects of music theory, such as key signatures, clefs, minor/major keys, sharps, flats, major/minor scales, jazz scales, modes and intervals.

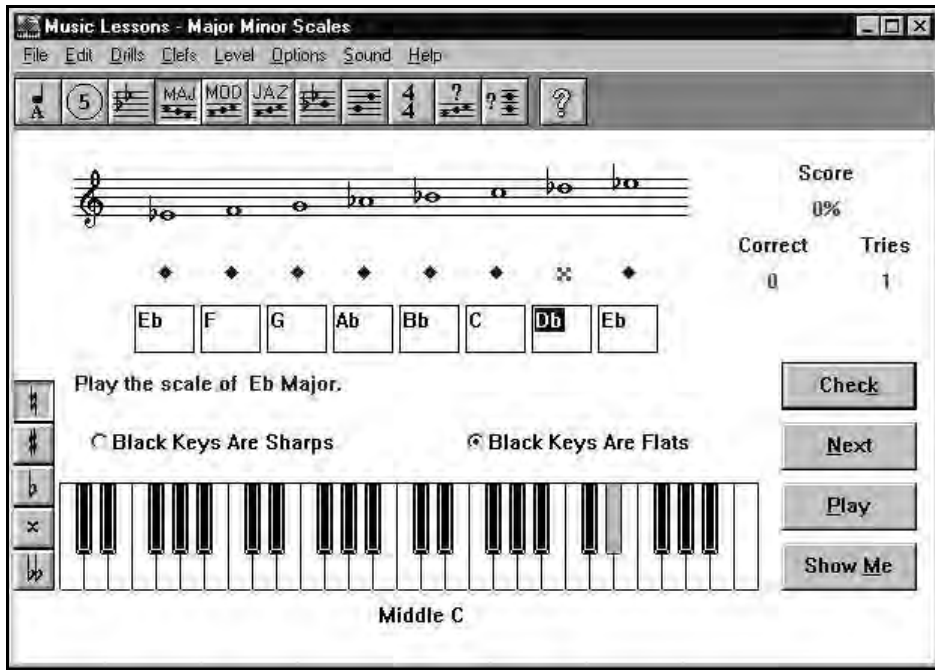


Figure 4: Screenshot of Music Lessons I

Music Lessons II is recommended for intermediate to advanced musicians. It covers chords and harmony at a more advanced level than a beginner would be able to cope with.

Another program that covers music theory is *Essentials of Music Theory* [Alfred]. It covers music concepts from the basics, and has exercises for practicing. This program includes some games, such as word search puzzles, flash cards and Note Naming Bingo. (These features may not appeal to all who want to learn how to play the keyboard).

Musition is also useful for learning the fundamentals of music theory. It includes many drill-based exercises for practicing, and enables the student or a music teacher to add other chords and Reference information. *Musition* covers topics in classical and jazz theory. Some of the topics include scales, intervals, instrument range, note reading, advanced clefs, key signatures, scale degrees, symbols, terms, musical concepts, chord recognition, meter recognition, rhythm notation and transposition.

2.3.4 Instrument Focused Programs

The last category of programs teaches particular instrument skills. The music theory covered by these programs is integrated with the lessons, and the practice exercises reinforce both the practical and the theory. Although programs that teach one how to play the keyboard are the most common, there are programs available for other instruments such as the guitar.

Musicware Piano [Musicware] is considered ⁵ one of the best programs for learning how to play the keyboard. It emphasises reading music and recognising the sounds of different notes, as well as developing a sense of rhythm. It has numerous exercises for each of the lessons and is better suited to the more dedicated and serious student who wants to learn how to play the keyboard competently. It includes a songbook that can be referred to for further practice when needed. By the end of all the courses, the user will have a good understanding of chord structures, including inversions, major, minor dominant chords as well as knowing how to play major and minor scales. *Musicware Piano* can supplement the input from a music teacher or it can be used independently by a music student.

Another good program that is useful when learning how to play the keyboard is *Piano Discovery* [Jump Music], which was developed by professional music teachers. It emphasises reading music and covers the fundamentals needed to play the keyboard. There are video clips demonstrating certain aspects of playing the keyboard and some arcade style games for practising. A 16-track MIDI sequencer is included for creating and editing music.

Teach Me Piano [Voyetra] covers a range of keyboard playing skills, including note reading in the treble and bass clefs, rhythm and timing, finger numbering and finger positions, key signatures, time signatures, scales and chords. It also includes a songbook with some popular songs to practice and a musical Reference guide. It also relies on video demonstration, illustrations and animations to enhance the lessons. A *Media Check* is included for troubleshooting sound and MIDI keyboard problems.

⁵ It is no longer supported but is still available to purchase.

2.4 Summary

The keyboard is often chosen by new music students as their preferred choice of musical instrument. The keyboard can easily be connected to the computer due to the development of the MIDI protocol, and there are various resources available to use for learning how to play the keyboard.

Learning to play the electronic keyboard does include learning about certain fundamentals of music theory and includes the ability to recognise music notes the relationship between these notes and the music keyboard.

There are various programs available to the student wishing to learn how to play the keyboard. These programs were grouped into four types of programs, general music programs, aural programs, music theory programs and keyboard programs.

General music programs include music references and utility programs. The reference programs offer a source of information on music and musical instruments while the utility programs assist a student in configuring the MIDI connections on the computer.

Aural programs can be used to develop the student's ability to recognise musical notes and play the notes.

Music theory programs cover a broad category of programs that deal with music theory. These programs are not aimed at a particular instrument although the keyboard is often used by the programs for demonstrations and exercises.

Instrument focused programs are aimed at teaching one how to play a particular instrument. There are a variety of programs aimed at teaching keyboard skills to appeal to different learning styles and age groups.

In spite of the advanced features of these programs, they are still limited in that the program does not deviate from what it is programmed to do. The computer programs available cannot offer all the benefits that are gained from attending a music lesson given by a professional music teacher.

2.5 Concluding Remarks

A student who uses one or more of the afore-mentioned programs is likely, with time and practise, to become proficient at playing the keyboard. All the programs mentioned have a different style and approach to teaching keyboard skills, and as such, there is no one best program. Anyone aspiring to learn how to play the keyboard will need to devote some time and attention to choosing a program that they prefer or a program that will appropriately meet their needs. A program that is highly regarded by music educators for its content and style might not appeal to a music student at all. The program a music student chooses, and uses successfully, is one that they feel comfortable using. Different learning styles appeal to different people and the type of program preferred can vary from a very theoretical program to one with animations and games.

Even if one has all the latest software installed on the most up-to-date computer, this will not replace the time and attention of a music teacher. A music program cannot adjust to the mood or temperament of the student nor indeed inspire and encourage the music pupil. The programs are also limited in what they can present to the user. They cannot branch off into topics that they are not programmed to present, such as those a student may find interesting or useful.

The ideal scenario would be to combine the strengths of the music programs with the input of skilled music teacher. By combining the teacher's skills and modern computer technology, the ideal music teaching system may be possible, and this idea will be addressed further in the following chapters.

3 Current Technology Applicable to Classroom Music Instruction

The previous chapter focused on some of the different tutorial programs available for learning how to play the keyboard. Anyone wishing to learn how to play the electronic keyboard can choose from a number of programs to suit their needs. However, although there are some very good programs available, they cannot replace the input of an experienced music teacher. Attending a tutored music lesson is generally considered the best way to learn how to play the keyboard [Zeigler]. The ideal situation combines the advantages of using computer music programs with the skills of a music teacher to make learning how to play the keyboard accessible to more people.

What is desirable for learning keyboard skills is for the student to have access to a music tutor who will give them individual time and attention while at the same time making full use of computer technology for the music lesson. How this can be achieved will be examined in this chapter by considering how computers and MIDI can be used for a keyboard class.

3.1 Limitations of One on One Music Lessons

Keyboard skills can be learnt by attending music lessons with tutored input or by an individual using a computer music program. Individuals seeking to learn how to play the keyboard can choose which method will suit them, but choice is dependent on a number of considerations, such as time, cost and availability of the lessons.

Attending a music class given by a music teacher can be expensive since the student is paying for the teacher's time and experience. Due to the nature of the music lessons, a music teacher normally only has one person at a time for a lesson. It is possible to have a small group of students for a music lesson, but this would be dependent on the students in the class being at a similar ability level [Spruce, 1996].

For the past fifty years, music has been taught to classes within the schooling system as 'band practice'. These classes consist of the music pupils, with their assigned instruments, and the band teacher who instructs the band as a whole, not the individual students [Spruce, 1996]. The emphasis is not on the individual students and their understanding and appreciation of

music. Those wanting to learn about music, and how to play a particular instrument, benefit most from a one to one session with a teacher [Zeigler].

Learning how to play the keyboard is a very practical undertaking and there is a lot of interaction required between the teacher and the student. The teacher not only demonstrates techniques used, but also watches and corrects a student's technique. A lesson of this kind is very teacher intensive. The demands of a class of music pupils on a teacher's time and attention can easily become overwhelming, leaving the teacher ineffective as a music teacher.

The alternative to the music lesson is to teach oneself how to play an instrument using books, videos or computer programs. Various programs have been discussed that can be used when learning how to play the keyboard. The advantage of these programs is that they enable a student to learn and explore the music in their own time and they offer plenty of opportunity for practice. Some programs include a wealth of information about the theory of music and different instruments. The better programs cover the topic of learning how to play the keyboard and the theory of music very well.

A major disadvantage of these programs is that they cannot respond to the individual needs of the student. Many programs were developed in such a way as to cover the important topics according to a fixed schedule, which does not allow for diversions. The programs cannot inspire a student or react to a student's temperament and learning style. The excitement of using a music program may not last long enough to motivate a student through difficult periods, or indeed encourage a lifelong interest in music.

In spite of all the benefits these programs have, they do not match the skills or personal input of an experienced music teacher. Computer programs available for learning how to play the keyboard cannot be a substitute for the interaction with and input of a music teacher.

3.2 Suggestions for Teaching a Class Keyboard Skills

Ideally, there need to be more music teachers offering music lessons at an affordable fee, however, the reality is that there are not more teachers offering the lessons and this scenario is unlikely to change [Midwest, 2003]. The answer lies in making better use of the teacher's time as well as enabling them to teach more students in a music class at the same time. By using technology that is currently available, this may be possible.

3.2.1 Computer Software

If a classroom is equipped with computers and the appropriate music software then the student can use the programs as practice tools. The students when practicing an exercise could use the computer music program. The computer program can be used to help a student to play the correct notes and at the correct tempo. The programs would be most effective for drill type exercises in the music class that would normally have required monitoring by the teacher for any mistakes. This would release the teacher from monitoring 'by rote' practising, and free up time to deal with issues particular to the various students. The teacher then has more time to spend with each student to encourage and inspire his musical interests.

There are also other benefits to using the computer in the classroom. The teacher can use a computer for the class administration and for tracking students' progress. Course notes and other materials can be made available to the students to use and copy so that they can be referred to later out of the lesson. The computer, if used correctly, can be a useful resource in the classroom situation [Spruce, 1996].

3.2.2 Networked Computers

If computers are used in the music classroom, there are practical advantages in having them connected to a network. Computers are easier to administer and maintain on a network, and the lessons can be delivered across the network from a central point to the students. Without the network, the teacher would have to copy the exercises individually for each student in the class.

A music teacher who is teaching keyboard skills to a class can use some of the programs discussed in the previous chapter. These programs allow for the distribution of tailored exercises to the class for use on their computers. The results of the exercises can be saved and sent back to the teacher to be graded and the results recorded. For example, *EarMaster* uses shared drives for transferring exercises and results, and allows the teacher to control the lessons [MidiTec].

Word processors, music software and multimedia programs can be used more effectively in a networked classroom. It is far easier to share the files across a network compared to physically copying the files for each student in the class. Other programs such as web browsers and email programs are also available in a networked class.

The World Wide Web (WWW or web) could also be deployed for teaching purposes. There are easy to use tools for web page creation. Course materials and exercises could be made available in a standard form for anyone in the class to view or save.

In The Laptop Project [Goddard, 1999] students for a music course were issued with laptops that had MS Office and Band-in-a-Box installed. Lessons were delivered as PowerPoint presentations. The teacher communicated with the students using email, NetMeeting and online chat sessions. Microsoft Access was used to keep track of the students' progress. It was found that the quality of teaching and learning was strengthened by the technology. Papanikolaou et al presented a paper titled "Deployment of Internet Resources for Digital Audio Learning Course" which focused on delivering Digital Audio lessons through the internet primarily using the web [Papanikolaou, 2000].

All the programs mentioned so far could be used for teaching a class music skills, however, the teacher cannot selectively monitor and interact with the students. A possible way for the teacher to monitor what the students are doing is to use a remote desktop program.

A remote desktop program enables the user to see and control the desktop of another computer on the same network. This includes controlling the mouse and keyboard of the remote computer. Programs such as *PCAnywhere* [Symantec], *Windows Remote Assistance* [Microsoft] and *VNC* [RealVNC] are used to connect to and control other computers. By using one of these programs, the teacher can remotely watch and control a student's computer. The teacher can also check on a student's progress with the exercises or configure their software, without disturbing the student.

A network of computers with the relevant software can be used to overcome some of the problems associated with teaching more than one person at a time how to play the keyboard. Some benefits of using a network of computers have been discussed, however the one critical area that limits the teacher in giving a music lesson to a class is the interaction between the teacher and student.

3.2.3 Specialised Education Systems

Specialised multimedia systems are available for classroom use and are specifically designed to run on a computer network. These teaching aids are not necessarily aimed at music or keyboard classes in particular. One such program is *WinSchool* [Chancery], which allows for the transmission of audio and video clips to each student in the class. In figure 5, there is a diagram of the teacher's display. Each student in the class appears in the display, and the various functions of the program are easily available for the teacher to use. This system relies on an extra set of cables running parallel to the computer network cables for the delivery of the audio and video data. Extra hardware is required to connect the multimedia cable to the computers. Systems like this tend to have high installation and maintenance costs.



Figure 5: WinSchool's class view

NetOpSchool, [CrossTec] seen in figure 6 is an administrative and teaching tool for use in a classroom. The teacher is able to monitor the student's computers and to see what they are doing on their computer. The teacher can also transmit his display to the class for demonstration purposes. *NetOpSchool* is very similar to *WinSchool* except that it does not require any extra cables or hardware.

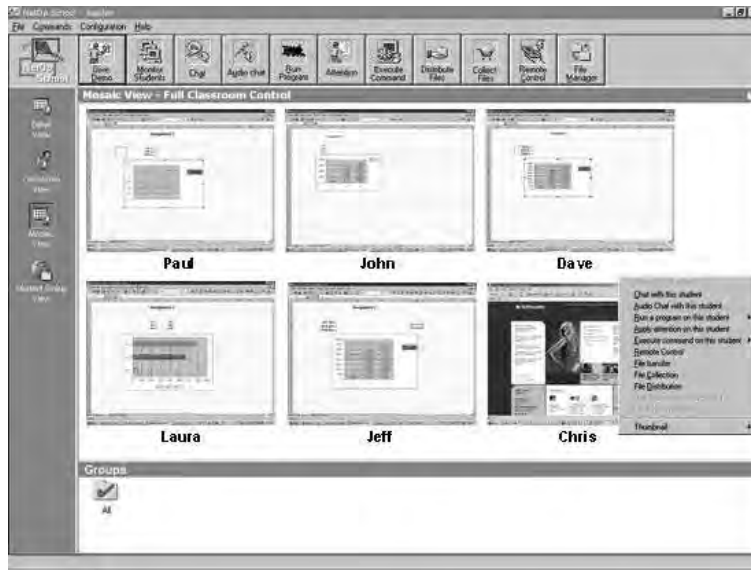


Figure 6: NetOpSchool teacher's view

Although both the programs have features that make class administration easier, they do not contain all the features necessary for optimising teacher/student interaction during the keyboard lesson. These two programs are aimed at administering a class and presenting computer based lessons. In a music class, the emphasis is on music and the teacher and student being able to hear and see what is happening. There is no possible way for the teacher to monitor what the student is playing on a keyboard. No use is made of MIDI for transmitting musical information between the programs.

3.2.4 Video Conferencing

The most effective way to optimise teacher/student interaction is to use video conferencing programs. These would allow the teacher and student to see and hear each other. Computers can now capture video in real-time and transmit the images to other computers on the network. There are some fast and effective compression and decompression programs available.

Video conferencing is now gaining popularity and the maturing standards are being adhered to with much success [Miller, 1998]. Video conferencing can overcome the shortage of teaching staff in some locations, as well as delivering specialised courses in a variety of locations [Independent Education, 2002]. Computer Science staff at Rhodes University present lectures to students at Fort Hare University using a video conferencing program

[Penton, 2000]. This method of delivering lectures is proving to be a very effective way of making limited resources (such as the teacher time and skills) available to a wider audience. Video conferencing is primarily used between two or more remote sites to overcome distance barriers, but there is no reason it cannot be used across a classroom network.

Video conferencing programs like *Microsoft's NetMeeting* [Microsoft] are relatively easy to use and freely available. They enable interaction between teacher and student, and include other features such as the ability to transfer and share files and applications, or have chat sessions. The teacher can assist the students while they practice by using the application sharing facilities, and at the same time, they can see and hear each other. This can be done without disturbing the others in the class.

The drawback of a video conferencing program is that it is typically designed for use in a one-to-one scenario. The teacher who needs to communicate with more than one student at a time in order to demonstrate something will find it difficult using a program such as *NetMeeting*. The teacher could set up a multipoint video conference, but this is a complicated task and requires extra hardware. In figure 7, a one-to-one and a multipoint video conference are illustrated.

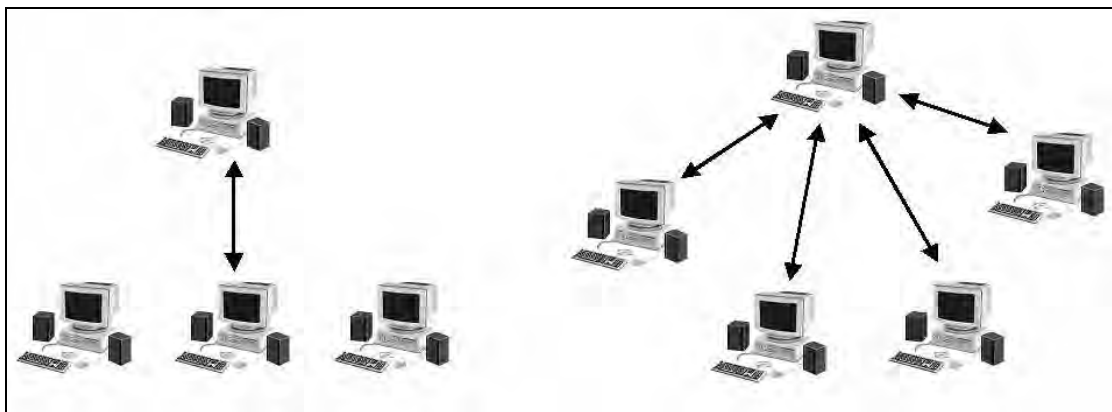


Figure 7: One to one and multipoint models

One-to-one conferencing requires that the video conferencing application be set up on each computer where it is used. For a multipoint conference, a ‘distributor’⁶ is needed on the teacher’s computer to control the communication channels between himself and each of the students in the class. It is possible to set this up, but the teacher needs to be technically competent, particularly with regard to installing and configuring the hardware.

⁶ Controls multiple input and output streams of multimedia data.

Video conferencing programs are mainly used to allow two people who are talking to each other to see each other. There are limitations in using these programs for a music class. These include the quality of the audio and video, and the synchronisation. In programs such as NetMeeting, the quality of the audio transmission is not a high priority and there can be up to a one-second delay in the transmission [Penton, 2000]. For a music lesson, where sound is important when demonstrating technique, synchronisation of the audio and video is critical for the lesson to be successful. In a normal video conference, it is not too distracting when the audio and video are not synchronised. In a keyboard lesson, it is confusing when the musical notes heard do not match up to the keys seen pressed. This is particularly confusing when a sequence of notes is played. In addition, the sounds produced in a keyboard lesson should ideally be heard at the highest audio quality possible. This is particularly important for the student who is still becoming accustomed to the pitches of the musical notes.

If the musical notes were transmitted as MIDI messages and not as audio data, then the musical notes could be played back at the highest quality possible from the synthesizer source. Video conferencing programs do not have this capability. Considering the benefits that MIDI has had for computer programs for teaching keyboard skills, it would be advantageous to include the transmission of MIDI along with the audio, video and other data in a video conference.

3.3 Summary

To combine the advantages of a tutor led music lesson with the available computer technology would allow a music teacher to effectively teach a class music skills. Music lessons tend to be expensive since the all the teacher's time is taken up by a single student, computer programs are very useful, however they cannot replace a music teacher. Various possibilities for using computer technology by a music teacher in a class have been examined.

By having a classroom equipped with computers and the relevant software does go a long way to allowing a teacher to teach a class music skills, but it does not provide an ideal solution. The teacher has to wander around the class and somehow listen to what the student is doing on the keyboard without disturbing the students in the class. No use is made of MIDI, which is still limited to the connection between the student's computer and his keyboard.

The use of dedicated classroom software can assist the teacher with classroom administration and presenting lessons. There is no capability to transmit MIDI between the teacher and students using these dedicated programs, and these systems often require extra hardware for audio and video transmission that adds complexity to the solution.

Video conferencing applications do offer the ability for a teacher to communicate with a student without disturbing others in the class. These applications are, however, not designed to be used in a classroom and, again no use is made of MIDI or the ability to share musical information. The communication style is more suited for use by a group of peers and not for the structured classroom situation where the teacher remains in charge of the communication.

None of the discussed options provide a particularly good solution for use in a music class. Each of the discussed possibilities do, to some extent, allow a teacher to effectively teach a music class keyboard skills. No use is made of MIDI and no musical information can be shared between the teacher and students other than the audible sounds.

3.4 Concluding Remarks

Technology can be an important and valuable resource in any educational setting. The appropriate use of technology in any lesson can enhance learning.

The use of audio cassette players, record players and compact disc players in music classes means that students can listen to pieces of music that support theoretical input. Teachers can easily operate technology of this kind and use it effectively within the classroom context. It is essential that any technology used within the classroom does not interfere with the teaching process. Technology must enrich the lessons without demanding the teacher's constant attention. The teacher must be able to give all his time and attention to the needs of his students. None of the abovementioned technologies present themselves as fully viable options even though they can be used to some extent to teach keyboard skills to a class.

While the technologies discussed in this chapter can assist students in learning to play the keyboard, they are not yet developed enough to be essential classroom aides. They require

specialist knowledge to set up and run, and they lack precision in terms of visual display and audio quality that is needed in a music lesson.

An idea of how video conferencing and MIDI can be combined and used in a system for teaching a class keyboard skills is further discussed in subsequent chapters.

4 An Interactive System for Teaching Keyboard Skills

Some of the resources available for teaching keyboard skills to a class were investigated in the previous chapter. It was suggested that the available resources could be used to supplement the input of an experienced music teacher in order for the student to benefit as much as possible from a music lesson.

Combining the benefits of the input of a music teacher and computer technology means that more students can have access to music lessons. The limitations of computer technology and the importance of teacher input were also considered. It is necessary to explore other options in order to bring the best of both worlds together.

The concept described and developed in this chapter is that of an interactive system for teaching keyboard skills to a class, in an attempt to overcome the limitations of what is currently available.

4.1 Requirements for an Interactive System for Teaching Keyboard Skills

In a typical music lesson, it is essential that the student interacts with the teacher in order to learn how to play a musical instrument. In a classroom, a teacher's time and attention is divided amongst the students, which means that an individual student may not fully benefit from the lesson.

Using computer technology, a teacher could engage with students in a lesson while focusing on one student at a time and seeing to individual needs as they arise.

4.1.1 Teacher/Student Interaction

There is considerable interaction that takes place between a teacher and a student during a music lesson. The teacher demonstrates various techniques, listens to and watches how the student performs during the lesson. The student, in turn, listens to and watches the teacher

demonstrate techniques, and then practices them. Communication is essential between the teacher and students.

Three types of communication models have been identified in the keyboard class. Firstly, the teacher could communicate with the entire class or a group of students in the class as illustrated in figure 8. This would involve each of the students being able to see and hear what the teacher is doing and saying and be able to hear what the teacher is playing on his keyboard. This would occur when the teacher is addressing the class and each student is listening.

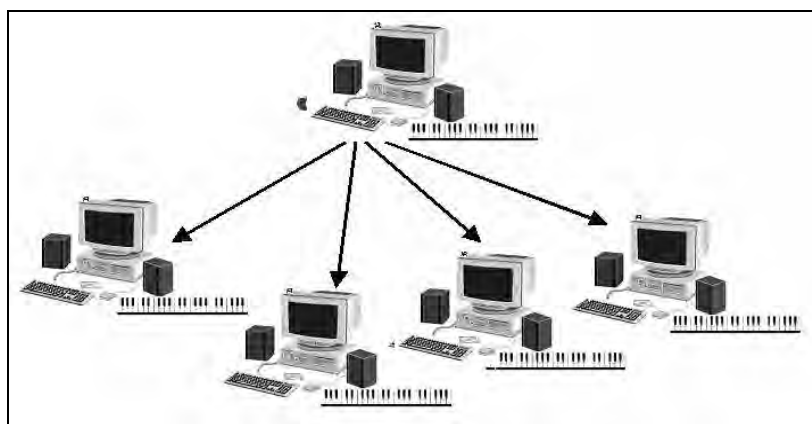


Figure 8: Teacher and class interaction

Secondly, the teacher and a student can communicate directly with each other. Both the teacher and student should be able to see and hear what the other is doing and saying. Figure 9 illustrates this one-to-one interaction. This situation can be compared to the teacher spending time with an individual student and assisting that student.

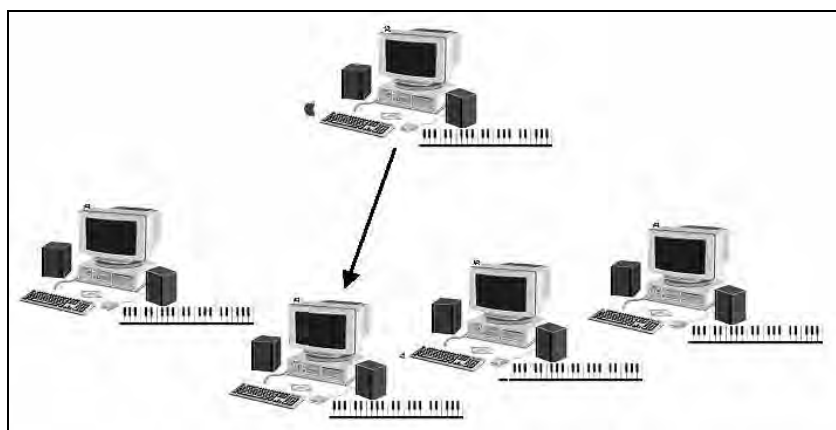


Figure 9: One-to-one interaction

Thirdly, in figure 10, the teacher can move from one student to the next, and monitor their progress as though he were wandering around the class and checking on each student. The teacher does not interrupt the student unless the student's technique needs correcting. Communication in this scenario is a one-way (from student to teacher) and non-verbal.

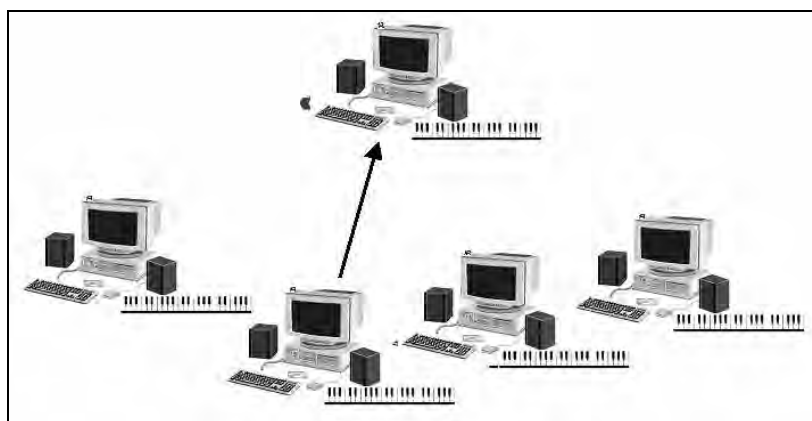


Figure 10: Teacher monitoring a student

For an interactive system to be effective in a music classroom, it needs to allow for all three types of communication between teacher and student, and deliver them in a high quality format across a computer network. This would involve transmitting multimedia data simultaneously between the teacher and student/s. This would provide the basis for an interactive system for teaching keyboard skills to a class.

4.1.2 Audio, Video and MIDI

Teacher and student need to see and hear one another in a music lesson. To see and hear involves the capture, transmission and playback of audio and video data. What is also needed for this system is the capture, transmission, and playback of MIDI data. The MIDI data would allow for the transmission of music information between the computer software and the keyboards. A teacher can then play a piece of music on his keyboard and the student would hear it as though it were being played on his keyboard. The audio, video and MIDI data is captured on the teacher's computer and sent across the network to the student's computer where the data is played back and displayed.

Since MIDI would be used for transmitting the music information and not be part of the audio data, the audio would not need to be high quality audio. The audio quality only has to be of sufficient quality so that the teacher and student can hear and understand what the other is saying. Since note information can be transmitted using MIDI data, it would mean that the music heard on the receiving end would be dependent on the device used to play the received MIDI data.

In an interactive music lesson, video quality is more important than audio quality (the transmission of the voices) since the music is transmitted using MIDI. The video images do not need to be in colour, but the frame rate should be high enough to catch the movement of a key being pressed and released on the keyboard. Bearing in mind that the system is intended use is in a music class for new students, not advanced keyboard players, the frame rate need not be too high either. Each computer used would display two video views – one showing the local camera view and the other showing the images received from the other person. The local video view would be used to position the camera and monitor what video images were being transmitted.

4.1.3 Music Teaching Software

Music teaching software can still be used to assist in teaching keyboard skills. As already discussed in chapter 2, there are some comprehensive programs available to teach keyboard skills. A music teacher can use these programs extensively in the music lesson, allowing the student/s time to do the exercises while monitoring each student's progress. The teacher can set exercises from these programs and upload them to the student's computers as required.

Any combination of the programs could be used for the keyboard lesson. The teacher could ask the student/s to use *Cakewalk* or some other sequencer to play a piece of music or even compose an original piece. Using the system the teacher can play a piece of music on his keyboard that is then recorded on the student's copy of the program for demonstration purposes. As the teacher plays the music, the MIDI information is sent to the student's program where it is played back through the student's keyboard and recorded. It is also possible for the teacher to receive MIDI data from the student as the student is practising, the teacher can hear exactly what the student is playing on the keyboard without disturbing the student or any others in the class. The teacher can develop exercises for the drill-based programs for the students to practice their keyboard skills.

An important part of learning how to play the keyboard is being able to use the current software with the MIDI keyboard. It may be necessary for the teacher to demonstrate to the class how to set up the necessary options so that the programs receive the MIDI messages from the keyboard. It should be possible for the teacher to then help individual students with any difficulties they may encounter using these programs.

4.1.4 Remote Desktop Sharing

Some of the features found in most video conferencing packages [Miller, 1998] are useful in an interactive teaching system in the classroom, such as remote control desktop capabilities and the ability to transfer files.

Exercises set by the teacher or MIDI files can be copied to the students' computers for them to use when practicing. This can be done in the classroom by using shared drives on the computers. It is not necessary to add any other mechanism for transferring files such as file transfer protocol (FTP) or a custom utility for copying files.

Remote desktop functionality would allow the teacher to control students' computers, configure the programs, or to demonstrate a music program to the students on their own computers. The teacher can see each student's desktop and the exercises or music they are currently playing.

4.1.5 Classroom Management

Classroom management is a necessary requirement for an interactive system of teaching keyboard skills as it is in any other classroom. Teachers need to know who is present in the class and have a means of controlling the class. There are two parts to classroom management in this interactive system - the class list and the assistance list. The class list would show all the names of the students in the class. As students join and leave the class, their names are added or removed from this list. The class list is the means by which communication is set up between the teacher and student. The assistance list would alert the teacher when one of the students in the class needs help.

4.1.5.1 Class List

A class list is necessary to log a student's attendance and control access to a music lesson. The class list could include a list of the students' names, the students' identification numbers or even the computer that they are using. For this system, the class list will use the students' names that they enter when starting their programs.

To join a music lesson, a student simply has to log in. This could be done automatically when the student starts their program, or the student may be required to fill in their name and possibly a password. The exact details required would depend on what is needed and most suitable for the music teacher. In this system, the students are only asked for their name when they start their program. The student's name is sent to the teacher's program and displayed in the class list. The teacher may then choose to deny a student access to the class, or even remove a student from a class.

The class list has two parts, namely the 'transmitting list' and the 'receiving list'. The two lists of names would appear identical to each other. Alongside each list there would be check boxes displaying the types of multimedia data that are available for transmitting and receiving. Figure 11 shows the two lists and their associated multimedia options. The transmitting list would be used when the teacher wants to transmit multimedia data to students. The receiving list would be used by the teacher to select from whom they want to receive multimedia data and the type of data he wants to receive.

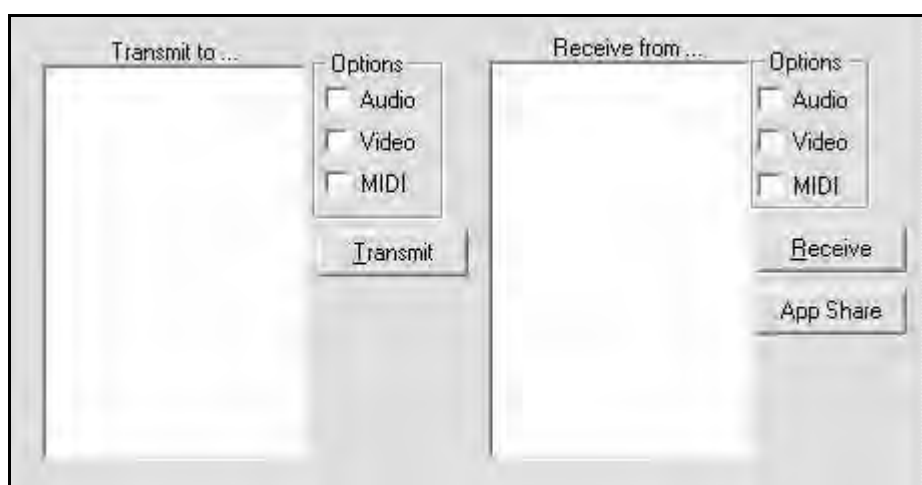


Figure 11: View of the class lists

Using the two lists, the teacher controls which of the three modes of communication, described in chapter 3, to use. The students in the class would not be able to communicate with each other using the system and they would not be aware, at least from the program perspective, how many students are in the music class. It would appear to each as though he were the only one in the class.

To broadcast to the whole class, the teacher would first select all the names in the transmission list and then select the type of multimedia data to transmit. The types of multimedia data that can be selected for transmission include audio, video and MIDI. After the names have been selected and the type of multimedia data chosen for the transmission, the teacher can then start transmitting the multimedia data to the class. If the teacher selects only a subset of the names from the list, then only those selected students would receive the transmission from the teacher.

If the teacher wanted to communicate with a single student in the class, then both the transmission list and the receiving list would be used. The teacher would select the student from the transmission list and the types of multimedia data to transmit. Next, the teacher would select the same student from the receiving list, and select the types of multimedia data to receive from the student. The teacher remains in control of the communication and is responsible for initiating and terminating all the types of communication. After the student has been selected in both lists and the multimedia data types are selected, then the teacher can start transmitting and receiving the multimedia data. The teacher and student will be communicating exclusively with each other, and will not disturb or affect the other students in the class.

The class list enables the teacher to communicate with a student. The receiving list enables the teacher to monitor a student. The teacher has the option of monitoring any combination of the available types of multimedia data, video, audio, MIDI or desktop view. When a student is being monitored, he will not be interrupted and thus not be aware that the teacher is watching. Depending on circumstances, it may be preferable to have an indicator on the student's program to show when the teacher is monitoring them. This would allow the student to ask questions or seek some advice or guidance from the teacher when being monitored.

4.1.5.2 Teacher Assistance

A student who needs assistance while logged into a music lesson will need some way of requesting the teacher's attention from within his own program. This will be possible by clicking a button that will notify the teacher that the student has asked for assistance. When the student clicks the button, his name appears in the assistance list on the teacher's program, as shown in figure 12 below. The students' names are added to the Assistance list in the order in which the teacher receives the request.

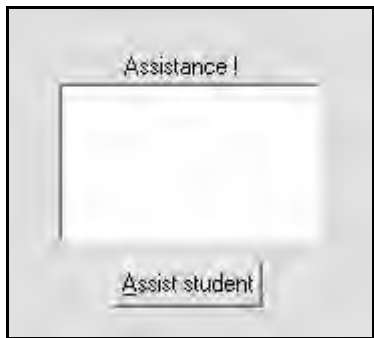


Figure 12: Assistance List

The teacher can then respond to a student's request by selecting a name from the list and clicking on the 'Assist student' button. This would then initiate a one-to-one connection between the teacher and student. Any other communication channels that were open would be closed. When the teacher starts to assist the student, that student's name is removed from the list.

4.2 Considerations for the Establishment of an Interactive System for Teaching Keyboard Skills

4.2.1 Ease of Use

It is important that an interactive system for teaching keyboard skills to a class be easy to use. The user interface should be uncomplicated displaying only the essentials of what is needed to use the system. The less cluttered the user interface the more intuitive and straightforward it will be to use [McKay, 1999]. The configuration options can be available through a popup menu so that it is easily accessible for any configuration changes needed, while not cluttering what the user sees initially.

4.2.1.1 Installation and Configuration

The installation and configuration must be straightforward, with as many default settings used as possible. The teacher using the system may not have enough technical knowledge to be able to choose all the correct multimedia settings. There must be clear and detailed explanations of the procedures followed and what the settings control. If using an operating system such as Windows 2000 or Windows XP, most of the set-up can be done automatically, and default inputs and outputs used. Use can be made of Wizards⁷ in the Windows operating system for the configuration of the audio, video and MIDI.

4.2.1.2 Class Use

The system is intended to be an aid for the teacher when delivering keyboard lessons to a music class. It is simply a delivery method and not a keyboard teaching system. It includes elements of video conferencing for use in the classroom.

The communication models are based on those used in a traditional classroom. The teacher is in control and the students cannot interact with each other while using the system. The details of exactly how the system would be used depend on the classroom layout, the teacher and the students in the class.

4.2.2 Availability/Cost of Resources

Most of the hardware that is needed, such as a MIDI keyboard and a computer with a sound card, would already be used by a student learning how to play the MIDI keyboard and may already be available in a music classroom. The extra requirements needed may include the network card, video camera, headphones and a microphone. All of the extra hardware needed for the system is easily available and affordable. There are no specialised items needed for this system.

⁷ Wizards or tasks that are available for configuring various aspects of Windows 2000 or Windows XP operating system and the peripheral devices.

Any video camera that is available as a 'web cam' would be sufficient for use in this system. The camera connects to the computer through its own dedicated video capture boards or through the parallel port or a Universal Serial Bus (USB) port. There is no need for an advanced high-resolution camera. Any set of headphones and a microphone can be used, provided they connect to the computer's sound card. The microphone would only be used for voice and not for the music, so again the quality of the microphone is not critical.

The network installed in the classroom would determine the choice of network card. Technical details of the hardware, including the networking details, will be discussed in the chapter covering the implementation of the system.

4.3 Summary

By combining the advantages of a tutored music lesson with computer technology, an ideal system for teaching a class keyboard skills has been described. Requirements for such a system were discussed along with other considerations that are important if it is to be used effectively in a classroom.

Teacher-student interaction during a music lesson is important. The teacher and students need to be able to communicate with each other, they should be able to see and hear what the other is saying and doing. The multimedia data transmitted between the computers includes audio, video, MIDI and remote desktop data.

By incorporating MIDI, the teacher is able to make better use of the connection between the keyboard and the computer and share music information with the students. The teacher is able to play his keyboard, and the student hears the music being played as if it was his own keyboard. Computer programs available for teaching music skills can also be better used by the teacher in the classroom in a similar manner.

A class list and assistance list allow the teacher to maintain control in the classroom. The teacher remains in control of all communication as in a traditional class situation. Students in the class can request help from the teacher using the assistance button on their program, the teacher can then respond to the requests for help.

Important considerations discussed included ease of use and installation of the system. The system should be available as a teaching aide and not detract from the music lesson.

4.4 Concluding Remarks

An interactive system for teaching keyboard skills to a class has been presented. A music teacher would use the system in a music class to teach the students keyboard skills.

The ideal system attempts to address the shortcomings of the options examined in the previous chapter. The extra features in this system include teacher/student interaction, the transmission of audio, video and MIDI, remote desktop capabilities, classroom management, and the use of music teaching software. All these requirements contribute towards making an interactive system possible for teaching keyboard skills.

An interactive system for teaching keyboard skills is the proposed solution that would allow a music teacher to effectively present a keyboard lesson to a class of students. What follows is the first step in developing a system to show that teaching keyboard skills to a class is feasible using an interactive music system. A design will be developed that will include all the features described for this ideal system, followed by the implementation of that design.

5 Design of a Remote Interactive Keyboard Teaching System

“The phrase “software design” means the conception, invention or contrivance of a scheme for turning a specification for a computer program into an operational program.” [McConnell, 1993] The design process is not deterministic as there is not a set of rules to be adhered to in order to produce a complete system design.

For small or simple programs, a good understanding of the problem may be all that is required in order to write the application. For the remote interactive keyboard teaching system, a design is necessary to clarify the details about how the system performs. The behaviour of the system is too complex to attempt to write the code without having a structured design in place.

The description of the remote interactive keyboard teaching system forms the starting point for the design. This ‘system’ consists of the programs and the necessary hardware, such as MIDI keyboards and video cameras. What follows is an overview of the approach used for the design including a discussion of the design as it was developed.

5.1 Design Approach

The Unified Modeling Language (UML) [OMG] was used for the design of the system. Grady Booch, James Rumbaugh and Ivar Jacobson developed UML in the mid 1990’s⁸. “The UML is the standard language for specifying, visualizing, constructing and documenting all the artefacts of a software system.” [Quatrani, 2001] It can be used throughout the development lifecycle from specification through to the implementation of a system.

UML consists of a number of graphical elements combined to form diagrams that represent the system. The different diagrams show various aspects of the system, from describing what

⁸ The UML specification is now maintained by the OMG and the latest release version 1.5 can be found on their web site <http://www.omg.org>

the system does to how it works. The advantage of a graphical design is that it can show the important aspects of a system quickly and easily and in an easy to understand way. UML is a modelling language, it is the notation for drawing the diagrams; it does not specify a process of how to use UML when designing a system [OMG].

The process followed for this design is based on the Unified Process as described by Booch et al [Jacobson, 1999] “The Unified Process is a development process. A software development process is the set of activities needed to transform a user’s requirements into a software system.” [Jacobson, 1999] The Unified Process is not a set of rules that have to be followed, but rather it is a set of guidelines that can be adapted to suite the current project. It is an incremental iterative process. As the design develops, previous parts of the design can be modified as necessary.

In the Unified Process, the main functions of the system are shown in a use-case model. This use-case model is the functional specification of the system. This model is central to the design, implementation and testing of the system.

5.2 System Specification and Description

A document describing the system and its intended use is the starting point of the design. A specification of the system describes how the program will work, who will use the program and how they will use the program. The description of the system in the previous chapter forms part of the inception and elaboration phase of the Unified Process. This stage is where the initial ideas for the system and the primary uses of the program are developed.

Details about what the program will do are described, but the focus is not on the technical details about how the program will function. For example, when transmitting data across the network from the teacher to a student, no mention is made about which network protocols are used to transmit the data.

Although the description of the system does not describe the technical details, these details can influence the description. In the description of the ideal system, the teacher receives multimedia data from the students. It was stated that the teacher first selects a student and then chooses the type of multimedia data to receive before receiving any of the data. The

function was described this way since it was known beforehand that to receive multimedia data from more than one source simultaneously is technically difficult.

5.3 Use-Case Model

A use-case model represents the behaviour of the system. It captures the main functions that the system performs as well as how the system interacts with its surroundings. The components of a use-case model are the actors and use cases.

Actors include anything or anyone that interacts with the system. They can be people, peripheral devices or other computer programs. A use case describes a fundamental function of the system. An example of this is the addition of a student to a class list when the student joins the class. “A use case is a sequence of transactions performed by a system that yields a measurable result of values for a particular actor.” [Quatrani, 1999]

Actors and use cases are combined in use case diagrams. These diagrams show the interaction between the actors for the various use cases and show who and what interacts with the system. “The collection of all the use cases for the system constitutes all the defined ways the system may be used.” [Fowler, 2003]

Some of the system’s behaviour, such as starting and ending the program, can be two separate use cases. Other possible use cases are a student joining and leaving the class, the teacher ending a lesson and the teacher or student transmitting and receiving the multimedia data. These are some of the possible functions of the system. However, before listing all the possible use cases it is necessary to find the actors.

5.3.1 Finding the Actors

The actors are not part of the system, they represent anything or anyone that interacts with the system. Either they use the system or the system uses them in some way.

The actors can be found by looking at the system specification and description for anyone or anything interacting with the system. The following questions were asked to help in identifying the actors for this system:

- Who is interested in a certain requirement?
- Who will benefit from using the system?
- Who will supply the system information?
- Who will receive information from the system?
- Does the system have external resources?
- Does the system interact with other systems?

[Quatrani, 1999]

Based on the answers to the above questions the following list of possible actors was compiled along with an explanation for their inclusion in the list:

- Teacher – uses the system.
- Student – uses the system.
- The teacher's program – sends data to and receives data from the student's program.
- The student's program – sends data to and receives data from the teacher's program.
- Video camera – the source of video data.
- MIDI keyboard – a source and destination for MIDI messages.
- Sound card – audio output, and source of audio data from a microphone.
- Graphics card – display and screen capture for remote desktop view.
- Network card – transmits data across the network and receives data from the network.

This list is not the final list of actors, as each possible actor needs to be examined to see if it is valid to be included as an actor for this design. By taking each possible actor, and looking at how it interacts with the system and what its purpose is, the final list of actors is found for the system.

- The teacher and student are treated as actors since they are the two main users of the system.
- The teacher's program and the student's program are part of the system under development. They are not external systems that interact with this system, they are therefore not included as actors.

Other possible actors are all hardware devices used by the system and treated as external systems.

- The video camera is a source of data for the system - it provides the video images for the system.
- The sound card provides audio data for the system and it receives audio data from the system to play.
- The MIDI keyboard generates MIDI message and plays the MIDI messages that it receives.
- The network card sends and receives data for the system. It is the connection between the two programs for the teacher and student.
- The graphics card is a source of data and it receives data from the system to display.

Based on the above examination of possible actors, the final list of actors is given in table 1:

Table 1: Actors for the design of the system

Teacher
Student
Network card
MIDI keyboard
Sound card
Graphics card
Video camera

5.3.2 Use Cases

“The use cases describe the functionality of the system available to the actors that produces an observable result.” [Fowler, 2003] Each use case captures a function of the system from the user’s point of view such as a student joining the class. The use cases define the functionality of the system.

To find the use cases for the system the following questions were asked as a guide to assist in finding the use cases -

- What are the tasks of the actor – what are the actors supposed to do with the system?
- Will the actor create, store or change any information in the system?

- Are there any use cases that will create or modify information in the system?
- Does the actor need to be informed about any changes in the system or any changes to the information stored in the system?

[Quatrani, 1999]

The above questions are a starting point for finding the use cases. Knowing ‘what the system is supposed to do’ and the ‘problem that needs to be solved’ are other sources for finding more of the use cases.

For this design, the system is separated into the two parts, the teacher’s part and the student’s part. The teacher and student are the two users of the system and they interact with the system in different ways. The system offers them differing functionality although the functions of the two parts are related to each other.

Starting with the main functionality that the system offers, as described in the previous chapter, the following is an initial list of use cases:

- Start the class
- End the class
- Join the class
- Leave the class
- Add a student
- Remove a student
- Ask assistance and offer assistance
- Configure communication options
- Transmit multimedia data
- Receive multimedia data

Most of the functions listed above behave differently for the teacher and student. A student starting the program and joining a class results in his name appearing in the class list on the teacher’s program. Likewise, when a student leaves the class his name is removed from the class list. The only use case that is unique to one part of the system is when the teacher starts his program; there is no corresponding use case on the student’s part of the system.

Looking at each of the use cases for the system listed above, and separating each one into the two parts produces the final list of use cases. There is a set of use cases for the teacher program and another set of use cases for the student program. All the use cases are listed in table 2 below.

Table 2: List of use cases for the system

<u>Teachers program</u>	<u>Students Program</u>
Start the class	-
Add a student	Join the class
Remove a student	Leave the class
End the class	End the class
Transmit multimedia	Receive multimedia
Receive multimedia	Transmit multimedia
Offer Assistance to a Student	Ask for assistance
Configure communication options	Configure communication options

5.3.3 The Use Case Diagrams

The use case diagram represents the system and the interaction of the actors with the system. It illustrates the main functionality that the system provides. The lines between the actors and the various use cases show interactions between the two. The actor generates events that trigger an action and produce a result in the system. Actors can also receive events from the system.

Using all the actors and use cases listed, the use case diagrams are drawn. Two diagrams for the two parts of the system show the interaction between the actors and the use cases. Figure 13 shows the interactions for the teacher's part of the system and the functions the system provides the teacher.

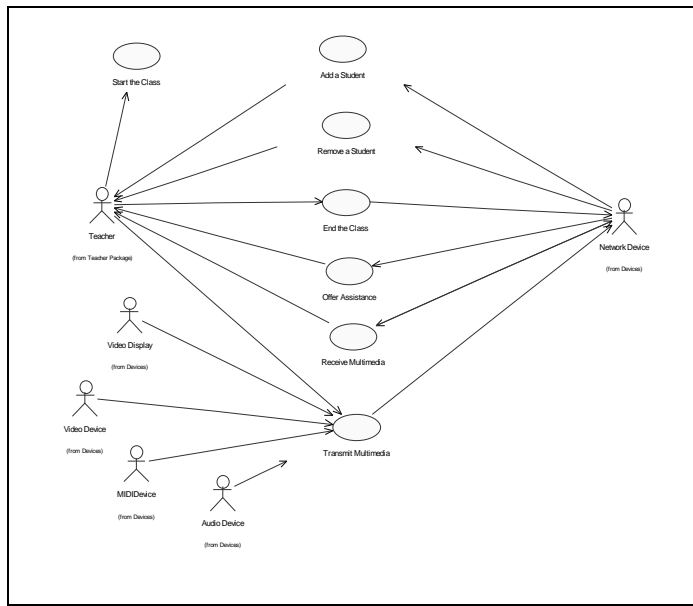


Figure 13: Teacher use case diagram

Figure 14 shows the interactions and functions available to the student.



Figure 14: Student use case diagram

The use case diagrams give an overview of what the system does and what is included as part of the system. It also shows who and what interacts with the system. There are no technical details about how to achieve the results.

5.4 Flow of Events

Each use case for the system has at least one scenario associated with it. “A scenario is a specific sequence of actions that illustrates the use case’s behaviour.” [Jacobson, 1999] The scenarios are described as a flow of events or as a textual scenario.

There are no clear guidelines for the number of scenarios needed for each use case, but there should be enough to give a clear idea of how the system will behave. The flow of events gives the details of how the use case starts, ends, and interacts with the actors for each use case. There should be enough detail to clarify how the system achieves the functionality offered to the actors. The sequence diagrams, shown later, are a graphical representation of these flows of events. The main purpose of the flow of events is to make sure that all possible behaviour in the system is handled correctly and documented.

A selection of the flow of events is given below and the rest of the descriptions are listed in the appendix.

5.4.1 Student: Join the Class

A student starts his program.

A window is displayed asking for the student’s name.

After entering his name, the program loads the audio and video preferences for the student.

The student’s name is sent across the network to the teacher’s program.

The student’s window is opened showing the local camera’s view and the remote camera’s view.

5.4.2 Teacher: Add a Student

A login request is received from the network.

The teachers program receives the data, and the new student's name and computer's address are added to the class list.

The new students name is added to the transmission list and the receiving list.

5.4.3 Student: Ask for Assistance

The student clicks on the 'Assistance' button to ask the teacher for assistance.

The students program receives the event and sends the assistance request to the teacher's program across the network.

5.4.4 Teacher: Offer Assistance to a Student

The 'Assistance Request' message is received by the network card from the network.

The message is passed to the teachers program from the network card.

The student's name is found in the class list.

The name of the student is added to the assistance list.

The teacher selects the student's name from in the assistance list and clicks 'Assist'.

The address of the student's computer is obtained from the class list.

The student's name is removed from the assistance list.

Multimedia communication is started between the teacher and student.

5.4.5 Teacher: Transmit Multimedia Data

The teacher selects a student from the transmission list.

The student is added to the list of students who will be receiving multimedia data.

The teacher then selects the type of multimedia data.

The teacher clicks the 'Transmit' button to start the data transmission.

The teacher's program sends a message to the network for the selected students, informing them what types of multimedia data to expect.

The teacher clicks the 'Stop' button to stop transmitting multimedia data to the selected students.

The message is passed to the teachers program.

The teacher's program sends the message to the selected students across the network.

The teacher's program stops capturing the selected multimedia types.

5.4.6 Student: Receive Multimedia Data

The network card on the student's computer receives the 'Receiving List'.

The message is passed to the student's program.

The student's program then checks the list to see if it needs to receive the multimedia that is to be sent and is prepared to process the selected types of data.

The network card receives multimedia data.

The data is passed to the student's program to be processed.

5.5 Classes and Objects

Having found the use cases and actors, and documented the flow of events, the next step of the Unified Process is to find the objects and classes of the system.

An object is defined as "An entity with a well-defined boundary and identity that encapsulates state and behaviour. State is represented by attributes and relationships; behaviour is represented by operations, methods and state machines. An object is an instance of a class." [OMG] It can represent a physical thing such as a MIDI keyboard, a music lesson or a list of students in the music class. When a set of objects can be described in terms of the same attributes, operations and relationship to other objects, they are considered to derive from the same class.

5.5.1 Finding the Objects

The objects for the system are found by examining the flow of events and extracting the nouns. [Larman, 2001] The nouns from each of the flows of events are added to a list, shown in table 3. Each noun is a potential object that will be considered shortly for its suitability as an object for this system. This is not the final list of all the objects in the system but rather it is a list of the potential objects.

Table 3: List of possible objects

Network card	Display	Video camera
Microphone	Speakers	Teachers' program
Student program	Music class	Keyboard
MIDI keyboard	MIDI events	Video data
Audio data	Teacher	Student
Multimedia data	Sound data	Login window
Button	Class	Video image
Class list	Mouse	Screen
Audio data	Sound	Network
Network data	Headphones	Graphics card
Sound	Video images	Video
Student name	IP Address	Check boxes
Multimedia	Remote video view	Local video view
Packets	Messages	

5.5.2 Filtering the List of Objects

Once all the nouns are listed, each one is considered in the context of the system's intended purpose [Larman, 2001].

Some of the nouns found refer to the same thing, such as video data and video images, while other nouns refer to more than one thing. Multimedia data in this system refers to audio, video, MIDI, and application sharing data. Other nouns are attributes of the objects. For example, student name and IP address are attributes of the student object.

The different nouns that refer to the same object are removed from the list and any nouns that refer to more than one thing are replaced by the individual nouns, when appropriate. It is necessary to have a good understanding of the system and its purpose when filtering the nouns so that any nouns used in place of others, are relevant to the system design.

Of the nouns listed in table 3, the noun in the first column of table 4 is kept in place of the rest of the nouns in the second column.

Table 4: Substituted nouns

Used Noun	Replaced Nouns
Audio data	Sound
Video data	Video images and video
Headphones	Speakers
Display	Screen
Multimedia data	Network data
MIDI data	MIDI events
Music class	Class, lesson and music lesson

Since multimedia data can be used interchangeably with audio, video, MIDI and remote desktop data it is not included in the final list. The teacher's program and the student's program refer to the two major components of the system and are not used as objects in the design.

The final step in filtering the nouns for the objects of the system involves looking at the type of objects they are and where they will be used in the design. In UML, various stereotypes are used. These include boundary class, entity class, control class [Jacobson, 1999]. For this design, only the entity class and boundary class are used.

Entity objects generally model information and behaviour that is long lived in the system, and not affected by how the system communicates with the surroundings. These objects do not have any direct contact with the actors, but they usually connect to the actors through other objects, including boundary objects [Jacobson, 1999].

After examining the list of nouns, the following met the above criteria and were labelled as entity objects:

- Teacher
- Student
- Class

The objects listed above will be labelled 'teacher unit', 'student unit' and 'class unit' when referred to in this design. This is to distinguish the teacher and student class from the teacher and student actors. The class is referred to as 'class unit' to avoid confusion when dealing with objects and classes.

Boundary objects model the communication between the actors and the system. Typical boundary objects include the program's window, communication interfaces and printer interfaces [Jacobson, 1999].

In this design, there is a boundary object for each actor. All the boundary objects are listed in table 5 below.

Table 5: Boundary objects

Actor	Boundary Object
Teacher	User Interface
Student	User Interface
Network card	System Interface
Audio card	System Interface
Video camera	System Interface
MIDI Keyboard	System Interface

The remaining nouns listed are not relevant for this design and are not considered further. Some of the entities, such as the keyboard and mouse and login window are incorporated into the boundary objects as part of the user interface. Similarly, the headphones are incorporated into the boundary class. The headphones could be replaced by speakers. Whether headphones or speakers are used does not affect the design.

The Unified Process attempts to separate the design from any implementation specific issues. The devices used for input such as the mouse and keyboard are regarded as part of the implementation stage and are not referred to explicitly in the design. The actual means of input is not critical to the system, a touch screen could replace the keyboard and mouse. The

login window is a window that is common to other programs, it is part of the user interface which is dealt with in the implementation chapter following.

The final list of objects used in the design of the interactive keyboard teaching system is listed below in table 6.

Table 6: List of objects for the interactive keyboard teaching system

Teacher unit
Student unit
Class unit
Teacher System Interface
Student System Interface
Network System Interface
Audio System Interface
Video System Interface
MIDI System Interface
Graphics System Interface (link to Teacher System Interface?)

5.5.3 Class Relationships

For this design, each of the objects listed is derived from its own class. Each of the objects offers methods that are unique to it, and how the objects relate to each other suggests they be derived from their own class and not a common class.

The relationships are found primarily from the flow of events. By looking at which objects are involved in each flow of events the relationship between the objects is determined. The relationship between the classes is represented as a line joining the two classes in the UML diagram. The two types of relationships, shown in figure 15 and figure 16, are associations and aggregations. Aggregations are the lines with the diamond on one end and the associations are the simple lines. An association indicates that one object knows about the other, or it uses the other object in some way.

Aggregation is a special type of relationship, it is a whole-part association. A student is a part of the class, this relationship is an aggregation. The numbers on the end of the lines

connecting the classes indicates how many of that class are involved in the relationship and is known as multiplicity.

For the teacher's part of the system the classes and their relation to each other is illustrated in figure 15. The Teacher Unit class is associated with each of the other classes and the Student class is part of the Class List. All the relationships are one to one except for the relationship between the Student and the Class Unit. The Class Unit consists of zero or more Students.

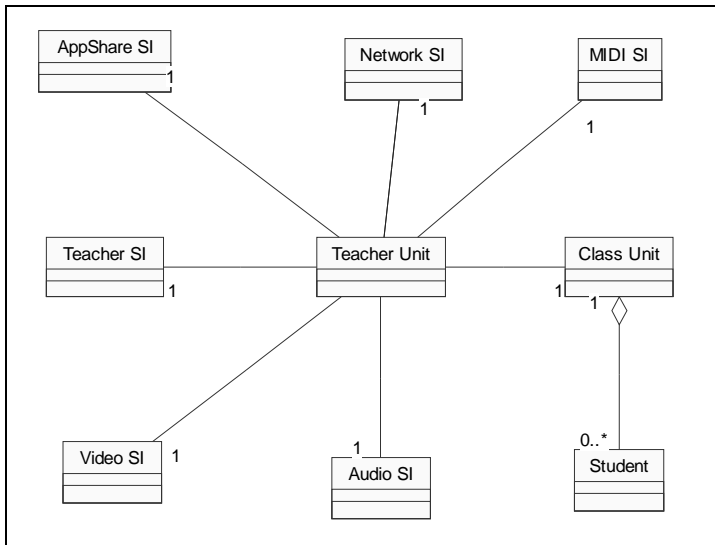


Figure 15: Relationships for the teacher side of the system

The relationships for the student's side of the system appear very similar to the teacher's side of the system, as can be seen in figure 16.

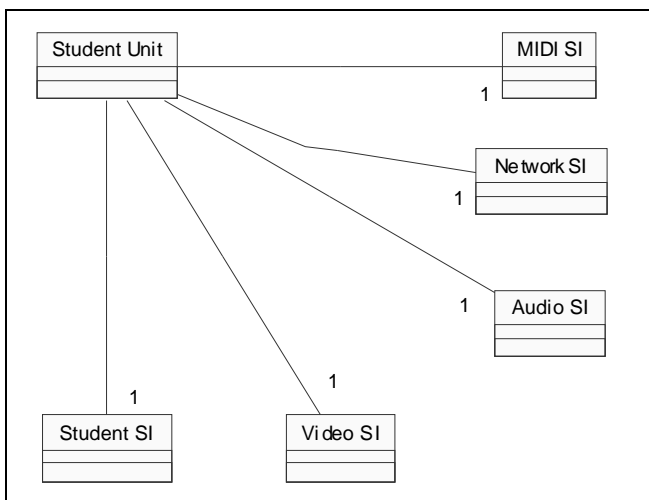


Figure 16: Relationships for the student part of the system

5.6 Sequence Diagrams

The use case diagrams show what the system does but they do not give any details about how the system behaves. Parts of the structure of the program are documented in the classes and objects and their relation to each other. The use case diagrams are static diagrams, they show the system and object interaction without any reference to time. What follows are diagrams showing how the system behaves in response to a user's request. They illustrate the flow of events for each scenario.

A sequence diagram shows object interactions arranged in time down the page while the objects are arranged across the top of the page. Messages indicate an object accessing a service of the target object. Messages are represented as lines with arrows pointing from the object that is calling the operation of the other object. The line label is the name of the service that is being accessed. These diagrams are most useful for developing an understanding of how the system works at a glance.

Sequence diagrams for the teacher and student parts of the system are related to one another, although for clarity they are treated separately. Three sequence diagrams, the student asking for assistance, the teacher receiving the request, and the teacher responding to the request, are examined in the following sections. The remaining diagrams are illustrated in the appendix B.

5.6.1 Student: Send Assistance Request

The sequence diagram in figure 17 shows the sequence of events that occur when a student asks for assistance. To ask for assistance the student clicks on the 'Assistance' button on his display. The student interface object handles the event generated by the student. The request is passed onto the student unit and then transmitted across the network through the network interface.

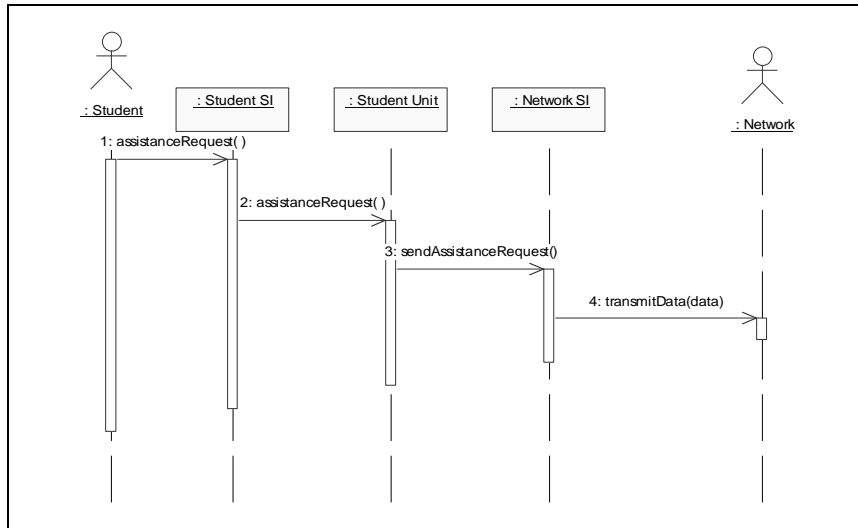


Figure 17: Sequence diagram showing student asking for assistance

5.6.2 Teacher: Assist a Student

An event is received from the network and it is passed on to the network interface. The network interface simply passes the event to the teacher unit along with the source IP address. The teacher unit fetches the students name from the list of students in the class using the source IP address as the index. Once the teacher unit has the student’s name it adds that name to the assistance list that is displayed on the teacher window.

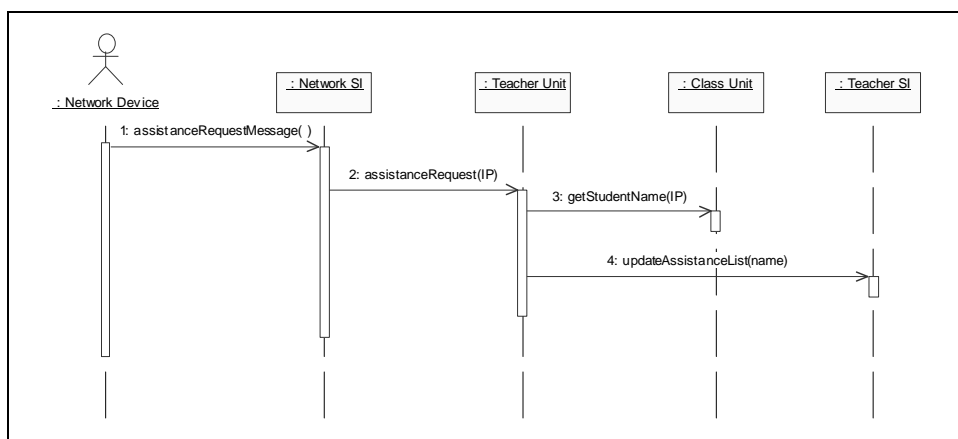


Figure 18: Sequence diagram showing the teacher receiving the assistance request

Figure 19 shows the teacher responding to a student who requested assistance. When the teacher chooses to respond to a request for assistance from a student, the teacher first selects

the student's name from the assistance list. The teacher interface receives the event and passes the student's name to the teacher unit. The teacher unit fetches the students IP address from the class list. The teacher unit then removes the student's name from the assistance list. Once the name is removed, the teacher unit triggers an event that sets up the communication options for a one to one session with the selected student.

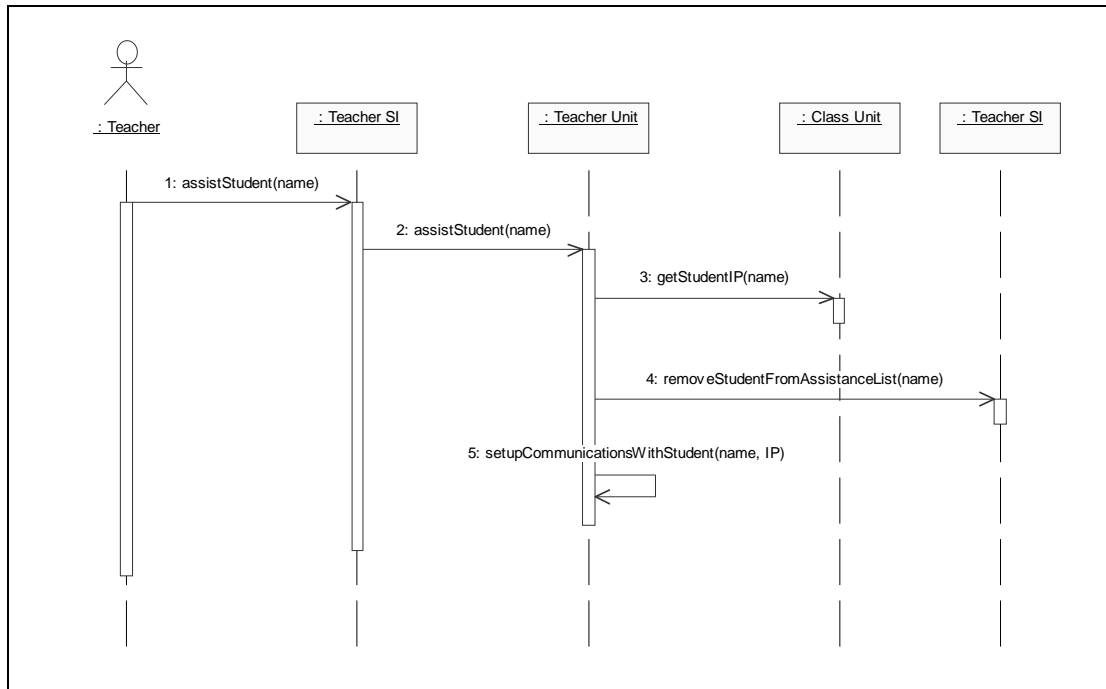


Figure 19: Sequence diagram showing the teacher responding to the students request for assistance

5.7 Summary

The Unified Modelling Language was used for the design of the system and the Unified Process formed the basis of the approach for the design. A description of the system and its specifications were used as a basis the design. Use cases describing the main functionality for the system were defined and the actors that use the system identified. Each use case had a scenario associated with it, which was described as a flow of events. Using the flow of events for each use case the classes and objects were extracted and defined. Finally, the sequence diagrams illustrated how the system would behave graphically.

5.8 Concluding Remarks

As the design progressed there were continual changes and modifications to the parts already completed, the Unified Process is an incremental and iterative process. What has been presented is the latest version of the design and the motivation for design decisions.

It is a high-level design that does not cover every detail of how the system behaves or how all classes are related to each other. The original purpose of the design was to clarify what the system does, how it should behave and act as a guide for the implementation of the remote keyboard teaching system. There is now enough information to direct the implementation of the system that is discussed in the next chapter.

6 Implementation of a Remote Interactive Keyboard Teaching System

The implementation of a remote interactive keyboard teaching system is discussed and certain implementation decisions are explained in this chapter. The design from the previous chapter is used as the basis for the implementation.

The implementation is intended to show that it is possible to have such a system implemented and it gives an idea of what it will look like and how the system will perform.

Moving from the design to the writing of the code is not a huge step, as the implementation should follow on naturally from the design. The biggest difficulties encountered were some of the technical details of working with the multimedia devices and capturing the multimedia data. The solutions to these difficulties and other implementation decisions will be explained.

6.1 Target Platform and Devices

The review of software for learning how to play the keyboard covered programs that run on the Microsoft Windows operating system. It therefore follows that this system will run on the same operating system since it is to be used in conjunction with some of those programs mentioned. Microsoft Windows does have a large share of the home computer operating system market [Statmarket]. This means that the largest possible audience have access to the system.

The hardware devices needed for the system have been mentioned in previous chapters. These included the sound card, video camera, microphone, keyboard and headphones. Standard devices that are compatible with Microsoft's Windows Operating System were used for the implementation. The devices are readily available and compatible with Microsoft Windows.

Any keyboard that supports general MIDI can be used for generating the MIDI messages. The keyboard the students use do not need to have any sound generating features since the MIDI notes will be played back using the soundcard. The teacher's keyboard will be used for

playing back MIDI notes received from a student, the explanation for this is covered later in this chapter.

Web cameras manufactured by Creative Labs were used during the development and testing of the system, but the system can use any video camera that is compatible with Microsoft's Video for Windows (VFW) drivers.

A 10 mega-bit Ethernet network was used to connect the computers together. This provided sufficient bandwidth for transmitting the multimedia data between the teacher and the students in the class. The performance of the network is discussed later.

6.2 Development Language and Tools

The language chosen for the implementation was Microsoft's Visual C++. This version of C++ was used primarily because of the Microsoft Foundation Classes (MFC), which simplify the development of Windows based programs and the fact that it is an object-oriented language.

The MFC are essentially wrapper classes for the Win32 Application Programming Interface (API). The MFC are also a framework that handle many of the routine tasks of a Windows program such as message passing and message handling [Prosise, 1999]. There are classes available that can be used as base classes for the windows, forms, buttons, lists and other objects mentioned in the design. All the details about how these objects behave and the messaging is handled by the MFC objects.

Accessing the hardware directly is possible using Microsoft's Visual C++. Windows Application Programming Interface (Win32 API) calls are easily implemented and there are APIs available for accessing all the hardware devices.

The Video for Windows Developer Kit , supplied by Microsoft, was used to capture the video and audio data that needed to be transmitted across the network. The developer's kit provides dynamic link libraries that can be incorporated into other applications. Using VFW, it is possible to view, capture, play back and edit video data. The Maximum MIDI Toolkit, supplied by Paul A. Messick [Messick, 1998] was used for the capture and playback of MIDI

messages. The MIDI messages were captured and processed using API calls made available through the Maximum MIDI Toolkit MIDI classes.

6.3 Graphical User Interface

The layout of the graphical user interface elements was described as part of the design in chapter 4. The user interface was used to illustrate the expected behaviour of the system. There is more to the GUI than how it appears, it can guide and restrict users in what they are able to do with the system.

The user interface for the teacher and student programs needed to be simple and intuitive. Since it is to be used as part of a music lesson, it must not be unnecessarily complicated and distract the teacher or student from the lesson. “User interface design creates an effective communication medium between the human and computer.” [Pressman, 2000] The following general principles were followed for the user interface –

- place user in control and
- hide technical details.

The student’s interface shown in figure 20 has only four items on it:

- A - the view of the student’s camera – the local video preview.
- D - the images being transmitted from the teacher’s camera – the remote video view.
- B - the “Assist” button.
- C - the “Leave” class button.

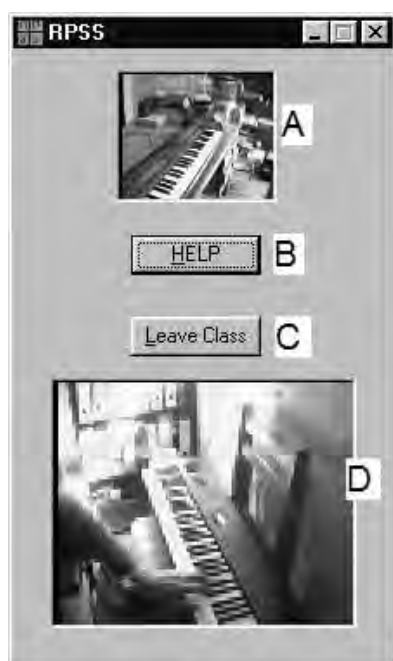


Figure 20: Student programs graphical interface

MFC Dialog boxes were used as the base class for both the teacher and student programs. The local video window, A, was based on classes provided by the VFW API. It uses a technique called video overlay to display the images captured by the video camera. Essentially, the image data is copied to the display's memory that is seen on the monitor. The teacher's camera view, labelled D, show the images that are captured by the teacher's camera. The image data is copied straight from the receiving socket to the video's memory to be displayed. The teachers program uses the same layout and techniques for displaying the local camera's images and images received from the student.

The teachers program, shown in figure 21, does have more items in the window since the teacher has the ability to control communication and respond to student's requests.

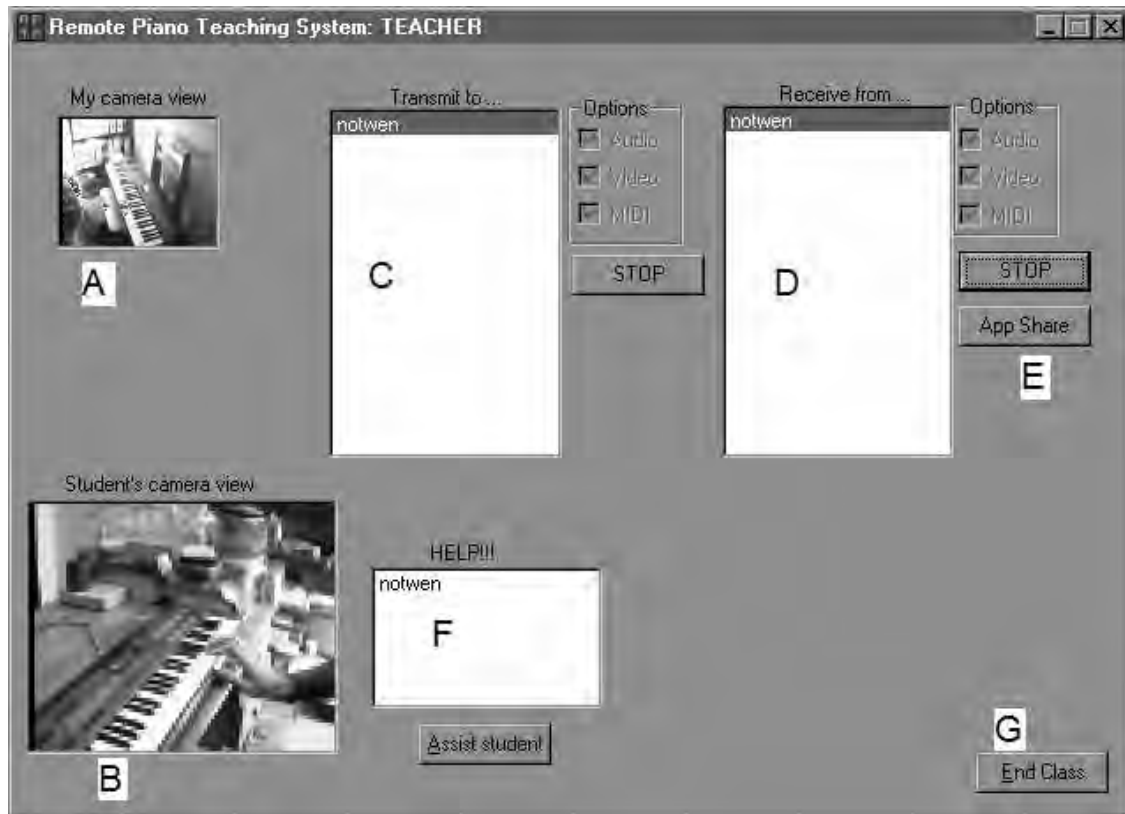


Figure 21: Teacher programs graphical interface

- A – Local video view
- B – Remote video view
- C – Transmission controls
- D – Receiving controls
- E – Application Sharing
- F – Assistance list
- G – ‘End Class’ button

Two identical lists are used for controlling the transmission and receiving of multimedia data instead of a single class list. Each student in the class appears on each of the lists. Before communication can be started, at least one name has to be selected from the list and at least one type of multimedia data needs to be selected.

The list on the left, C, is for selecting which students are to receive multimedia transmission from the teacher. More than one student can be selected from this list. The teacher can transmit to the whole class, a single student or a group of students. After selecting the name or names, the type of communication is chosen. It can be any one of the following: audio,

video or MIDI. Once the name or names have been selected and the type of multimedia data to be transmitted is selected the teacher can start transmitting the data across the network.

Only one name can be selected at a time from the receiving list, D, the teacher can only see and hear what a single student is doing and not a group of students. It is not possible to receive multimedia information from more than a single student at a time. Any number of names can be selected from the transmission list and all those selected will be able to receive any multimedia data the teacher transmits across the network. After selecting which student the teacher wishes to receive data from, the type of multimedia data is selected and then the teacher can start receiving that data. It can be any combination of audio, video, MIDI or application sharing data.

Application sharing is one of the types of multimedia data the teacher can receive from a student in the class. It is controlled in the same way as the other types of multimedia data, but the teacher cannot share his application with any of the students. The application sharing is discussed further in section 6.5.4.

The assistance list, F, is used when a student asks for the teacher's attention. If a student clicks on the 'Help' button, his name is added to the 'Assistance List'. The order that their requests are received is shown in the lists, the first received request is at the top of the list and subsequent requests are added to the end of the list. To make sure the teacher is aware that someone has asked for assistance a message box appears with the student's name on the screen. This was done so that if the teacher were busy with another application or not watching the screen he would either see or hear that a request had arrived. To respond to a request for help the teacher selects the student he wishes to assist and clicks the 'Assist' button. Once the button is pressed all other data that is being transmitted or being received is stopped. Audio video and MIDI data will then be transmitted to the student and received from the student.

No menus were implemented and all the audio and video settings were set in the program to use the windows default settings. This kept the user interfaces as simple as possible. These settings could still be available by using a hidden menu on the teachers program only. The student's program uses the default settings or the settings received from the teacher.

6.4 Network Protocols

The network protocols and format of the data were largely determined by the available network and protocols used. 10 Mb Ethernet and Internet Protocol (IP) were used in the laboratory where the system was implemented. Other considerations taken into account were performance issues and the intended use of the program. Speed and synchronisation are two very important factors for the audio and video transmissions. When talking with another person, long delays can be very disruptive to the conversation, and if what is seen and heard do not happen at the same time would also be disruptive. Another factor affecting the choice of protocol was the fact that the system is to be used on a LAN in a classroom.

User Datagram Protocol or (UDP) was chosen as the protocol rather than Transmission Control Protocol over Internet Protocol (TCP/IP). Implementing the system using UDP was simpler since connections do not need to be made and maintained by the programs. All the computers for the classroom are on the same subnet on a LAN so there are no network devices to affect traffic and visibility of the computers.

UDP is a connectionless protocol, and packet delivery is not guaranteed, packets can get lost. There are a very small percentage of packets that can get lost on a LAN [Microsoft Press, 1997] when using UDP. The percentage loss is acceptable for this implementation to demonstrate the feasibility of the design. Broadcasts are used to transmit data from the teacher to the student and can be done using UDP.

The amount of data that can be transmitted in a packet on the network was affected by the structure of the Ethernet frame and the structure of the UDP packets. Figure 22 shows these structures. The amount of data in each UDP packet was limited to 1492 bytes, which allowed a single UDP packet to be transmitted in a single Ethernet frame.

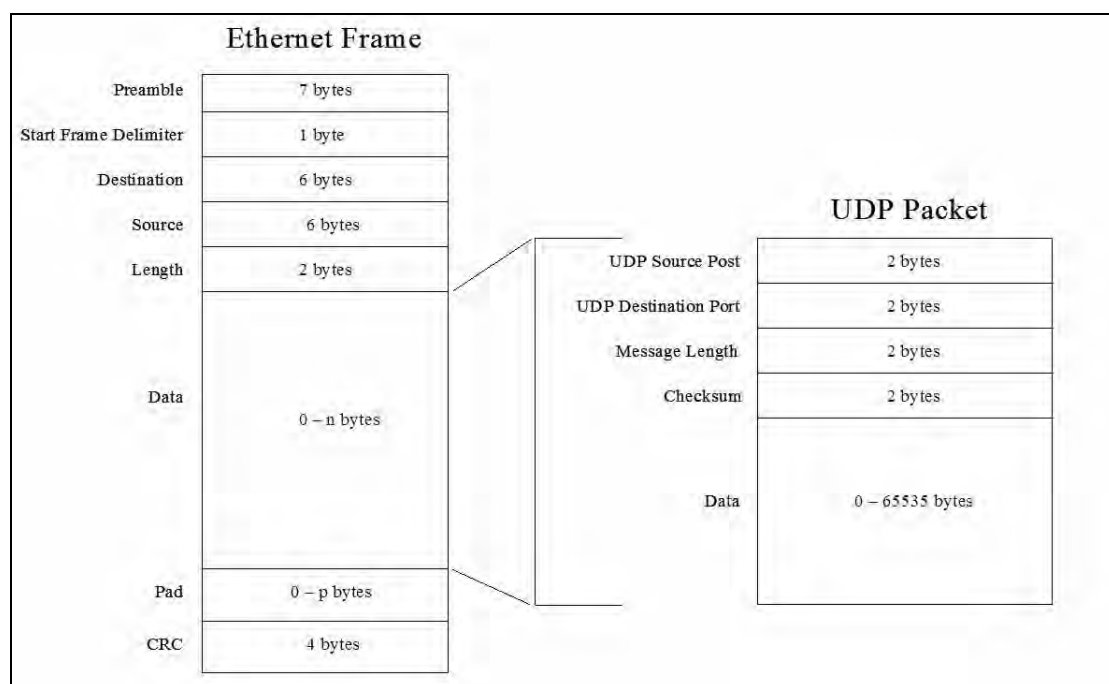


Figure 22: Ethernet and UDP packet structures

This was done so that the multimedia data arrived at the fastest rate possible and at a steady rate. UDP packets could be much larger and be transported using multiple Ethernet packets. This would mean that a lot more time was spent waiting for enough data to arrive to fill a UDP packet before it was sent. The network data, once it arrived, would be processed immediately and the UDP packet built up and passed on to the application. The application would then have a few seconds of data that needed to be buffered. This introduced delays and led to synchronisation problems. This issue is discussed further in Section 6.7.3.

In this program, the multimedia data was broadcast across the network. This is possible on the network in a classroom since the traffic can be restricted to that segment. All the computers receive the data but only those who were selected to receive the data process it. This follows closely the design of the system, where the teacher first selects students to receive data before transmitting any of the multimedia data.

Separate ports were used for audio, video and MIDI. The ports used were:

Table 7: UDP ports used for the multimedia data

<u>Type of Data</u>	<u>Teacher Sends To</u>	<u>Student Sends To</u>
Video	2071	2061
Audio	2073	2063
Control	2075	2065
MIDI	2079	2069

Each type of multimedia data has a set of associated ports, one for sending and the other for receiving data. The control data includes the student joining or leaving the class, the teacher ending the class and the teacher selecting which students are to receive any data transmitted from the teacher. When data arrives at a particular port, a callback function is called to process the data that arrives.

The address of the teacher's computer is stored in the HOSTS file on each student's computer. The student's program finds the IP address of the host name 'teacher' in the file. The host name of the teacher's computer does not need to be changed for the system to work. The teacher's program discovers the IP addresses of a student's computer when the student joins the class. The sending IP address is part of the packet that the teacher receives.

It would be possible to use TCP/IP and multicasting for the system. A socket connection could be used for the student notifying the teacher when he joins or leaves the class and for the teacher to instruct the students program to send multimedia data. This would ensure that the requests and notifications are received by the recipients. When the teacher wanted to transmit multimedia data to the class, he could transmit the data using a technique called multicasting. "Multicast is the sending of data from one to many recipients and would replace the UDP broadcasts in the system." [Miller, 1998] Depending on the destination IP address, only certain IP addresses in the multicast group would receive the packets for processing. This would be far more reliable for a commercial system, but it was not used for demonstrating a working system based on the design since it adds complexity to the implementation without adding any noticeable benefits to the prototype.

6.5 Multimedia Transmission

The formats chosen for the multimedia data are discussed in this section. The audio and video data can be represented using a number of formats such as PCM, mpeg, RLE and bitmap [Heath, 1999] formats. As well as the different formats, the quality and size of the data do vary. Choosing a format to use for this system was determined by the available codecs⁹, network bandwidth and processor time, which are discussed later in this chapter.

6.5.1 Audio

The capture and playback of audio is used in this system so that the teacher and students can hear each other speaking. No music is transmitted as audio data between the two, the music is transmitted as MIDI data and is discussed in section 6.5.3. Since the audio that is to be transmitted is to be used for voices the quality does not need to be very high and ‘telephone’ quality is used. ‘Telephone’ quality is the minimum accepted quality of sound for a conversation and has the following attributes shown in table 8: [Alvear, 1998]

Table 8: Attributes of telephone quality audio

<u>Sample Rate</u>	<u>Bits</u>	<u>Channels</u>	<u>Data Rate</u>
11 kHz	8	Mono	662 kB/min

The Video for Windows Developers Kit provides functions to capture and play back audio data. It uses the audio codecs that are provided by the operating system. The most common format is the ‘pulse coded modulation’ (PCM) format. A sound (waveform) is sampled at a set frequency and the amplitude is measured. The hardware converts the amplitude into a number in an analogue-to-digital converter (ADC). The data is converted back into a waveform using a digital-to-analogue converter (DAC).

Using the VFW, a callback function is set up. When the audio device fills its buffer, the callback function is called. This function copies the audio buffer to the network card and sends the data.

⁹ codec – COder/DECoder of data.

Once the audio data arrives at the destination a play function is called and plays the sound back through the computer's sound card. To get a smooth sound played back, a buffer is used for the audio data. Without using a buffer, the audio data when played back was broken and not continuous. This was caused by the fact that the audio data needs to be a continuous stream. Without the buffer, the sound is played back in chunks as the network packets deliver a slice of the sound stream. The audio data is copied to the buffer when it is received from the network. Each packet of audio data is copied to a section of the buffer as it arrives. The play function fetches the audio data from the buffer, plays the sound and then fetches the next slice of audio data from the buffer. This is illustrated in figure 23 below.

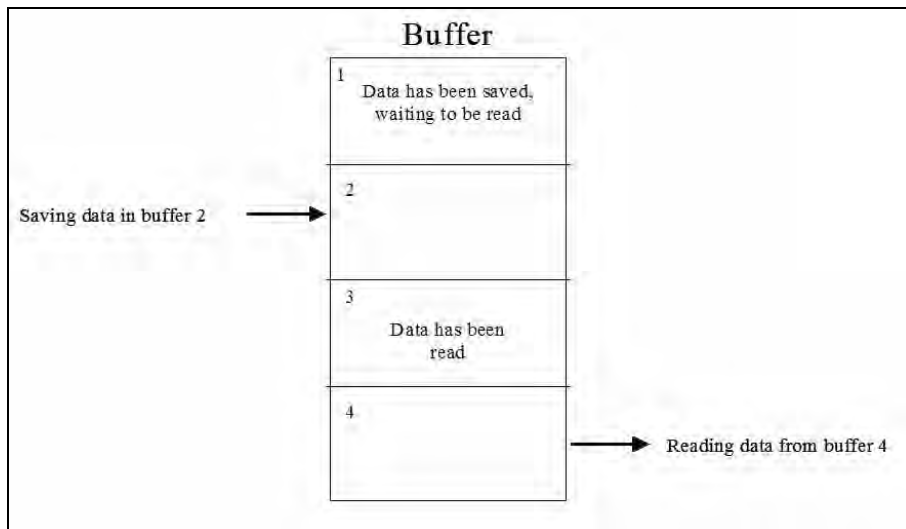


Figure 23: Audio buffers

The audio data is not compressed, it is sent as an uncompressed wave data. Compression would have added another delay in the capture and playback of audio data. The bandwidth available for this system was sufficient to handle the uncompressed audio stream.

6.5.2 Video

The video images are used in the system so that the teacher can see what the student is doing while playing the keyboard. Unlike the audio data that only had to be captured and transmitted, the video data also needed to be used locally for the local camera view. There are two video images on each of the programs. The local cameras view and, if receiving, the

remote camera's view. Video images can be sent and received at the same time and run independently of one another.

A frame rate of 15 was used for this system. This value was decided on by experimenting with differing frame rates as a beginner practices playing the keyboard and determining the effectiveness of the video image. Faster rates did not add enough to the video to justify the extra bandwidth used and a frame rate any slower meant that some finger movements were not seen.

The other factors affecting the quality of the video images are the colour depth and the resolution. The resolution of the images must be high enough so that the students' and teacher's fingers can be seen. The third factor affecting the video is the colour depth of the images. For this implementation, 88 by 72 images with 24-bit colour depth were used. The quality of the video images cannot be too high otherwise, too much data is generated and it is not possible to transmit all the data to the remote computer. These limitations become less of an issue more bandwidth is available and computers that are more powerful are used to capture and play the video images. These limitations are examined further in section 6.7.

The Video for Windows Developers Kit was used to capture the video images. VFW is primarily designed to work with Microsoft's Audio Video Interleave (AVI) file format [Microsoft]. There are functions for creating these files and saving video images to these files. The same functions were used for the capture of the video images but they were not saved to file, the video buffer was instead copied and sent in a UDP packet across the network. Callback functions were used for capture of the images.

The format for the video is received by the student's computer when he receives an instruction to process multimedia data from the teacher. The teacher's video format is the default setting for the system. All the members of the class use the same video settings.

Once the video images are received, the data is double buffered in a similar technique already described for the audio buffers. The one buffer is copied to the display device's memory while the other is being filled up with the received video data. The 'bitBLT' function was used to copy the data in the buffer to the displays memory without having to specify the format of the data. The previous format settings are used for each new set of data that is to be displayed.

The images that are captured are sent as bitmap images, this does consume a lot of bandwidth. For the purpose of this implementation, to show the feasibility of the system, it was decided not to implement any compression since adding compression would not have affected how the system looked and felt.

6.5.3 MIDI

MIDI messages in this system can come from more than one device. MIDI can be generated from the attached keyboard or from a MIDI program installed on the computer. Once the MIDI message has been generated, it can be used locally and sent to another computer. Any MIDI messages received from another computer can be sent to more than one device. It can be played back through the sound card, sent to a MIDI program or played back using the keyboard.

For the purpose of this section, the following configuration for the students and teacher will be discussed to illustrate how MIDI is used in the system. Figure 24 below represents the configuration for the teachers set up.

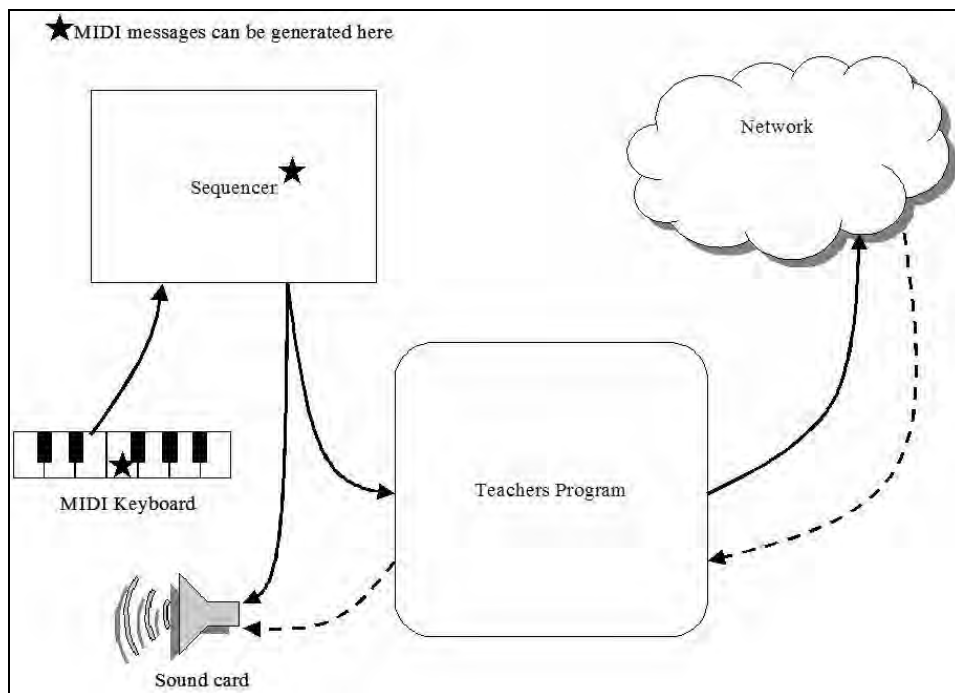


Figure 24: Source of MIDI messages on the teacher's side

The teacher is running a sequencer and the teacher's part of the system, RPTS¹⁰. The diagram shows where MIDI messages can be generated locally and where the MIDI messages are sent. The dashed lines in figure 24 shows the path MIDI messages take when they are sent from a student to the teacher.

The student side of the system is simpler than the teacher's side. Figure 25 illustrates how MIDI messages that are generated locally are routed. MIDI messages can be generated by the keyboard or the sequencer. The dashed line shows where the MIDI messages the student receives are routed.

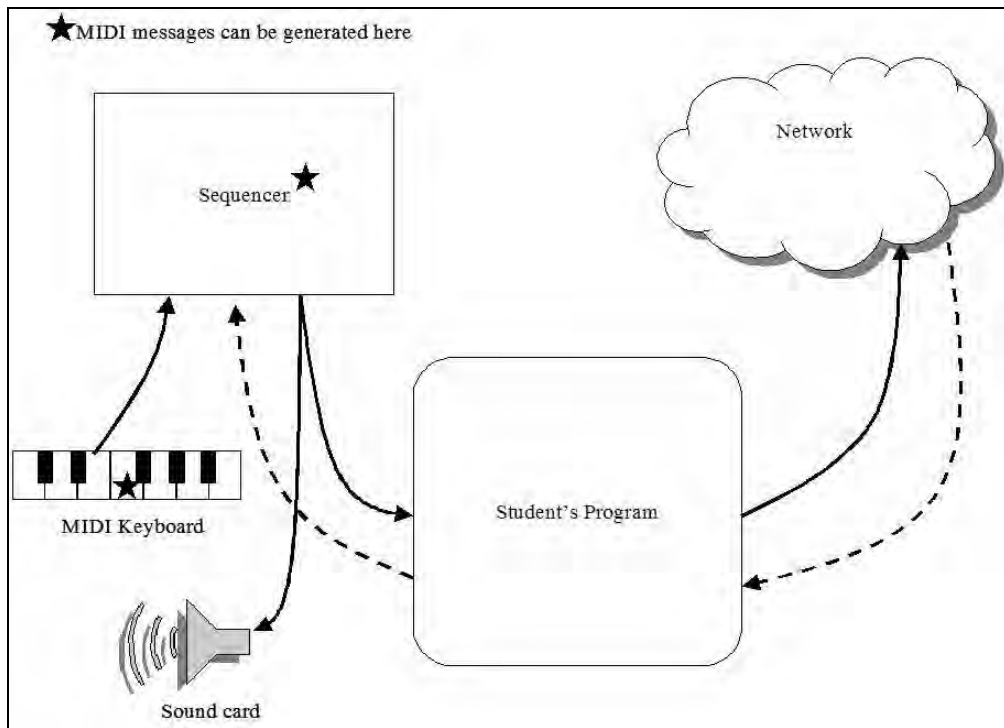


Figure 25: Source of MIDI messages on the student's side

MIDI drivers provided by Microsoft do not allow the MIDI messages to be sent to more than one port. This did limit the usefulness of the MIDI in the system. To achieve the above configuration a third party product called MIDIYoke [MIDIYoke] was used to route the MIDI messages to multiple devices from a single MIDI port.

MIDIYoke is comparable to a patch bay for MIDI. The output of two devices can be connected to a single MIDIYoke port. More than one application can listen to that

¹⁰ RPTS - The part of the system used by the teacher. Remote Piano Teacher Side

MIDIYoke port for MIDI messages. This allowed the student to play his keyboard, hear the notes being played through a sequencer and at the same time, the MIDI messages were captured and transmitted to the teacher.

Figure 26 illustrates where MIDIYoke is used for the teachers part of the system for the above configuration.

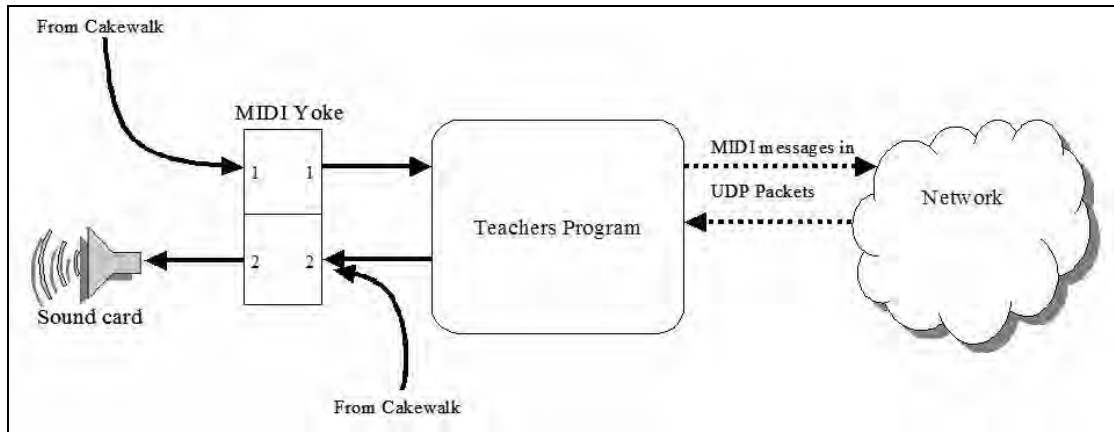


Figure 26: MIDIYoke connections

The teachers program sends and receives MIDI messages in UDP packets to and from the network. The teacher's program reads the MIDI message and sends the MIDI message to MIDIYoke Port 2. MIDIYoke Port 2 is also receiving MIDI messages from Cakewalks MIDI out. Port 2 In is mapped to MIDIYoke Port 2 Out that is passing the MIDI messages to the sound card where the note is played. MIDIYoke Port 2 In is receiving MIDI messages from two devices, Cakewalk and the teacher's program.

MIDIYoke Port 1 In is receiving MIDI messages only from Cakewalk. These messages are passed on to Port 1 Out. The teacher's program is configured to receive MIDI messages from MIDIYoke Port 1 Out. The MIDI messages are copied into a UDP packet and transmitted across the network.

This is not the only possible configuration but it does demonstrate that the MIDI can be captured and transmitted across the network for use in another program. The teacher can hear what the student is playing on the keyboard or what the student's copy of Cakewalk is playing. Using MIDI also allows the teacher to play a piece of music for the student to record in his sequencer as the teacher plays it.

6.5.4 Application Sharing

For application sharing, a program called VNC [Real VNC] was used. VNC was developed by ORL and stands for Virtual Network Computing, and it essentially allows you to view a computers desktop from another computer across a network.

Each student computer in the classroom has a copy of the VNC Server software installed. The server runs as a service and is started automatically when the computer starts up. The teacher runs a client program or can use Internet Explorer to connect to the students VNC Server. Once connected the teacher can see what is being displayed on the students monitor. It is possible for the teacher to control the student's mouse and type characters that will appear in the students program.

The VNC client is started when the teacher requests to receive application-sharing data from the student's computer. The VNC Server sends a copy of the display to the client program and passes on any mouse or keyboard events to the student's computer.

6.6 Classroom Management

In this implementation, 'classroom management' consists of two areas. The first is concerned with controlling the class list and students joining and leaving the class, and the second is controlling the multimedia transmissions between the teacher and student.

The control packets sent from the student to the teacher have the following structure, table 9.

Table 9: Student's control packet structure

<u>Description</u>	<u>Length</u>
Packet Type	6 bytes
Address	15 bytes
Name	30 bytes

The packet type can be any one of the following:

- Login
- Quit
- Help

The teacher on receiving a control packet from the student will remove a student from the class if a 'Quit' packet is received, add the student to the class if a login packet is received or add the student to the assistance list if a 'Help' packet type is received.

The second type of control packet is sent from the teacher to a particular student. If more than one student is to receive the packet, it is transmitted to each of the students consecutively. The packet consists of three characters for controlling the video, audio and MIDI in that order as shown in table 10.

Table 10: Teachers control packet structure

Byte number	1	2	3
Controls	Video	Audio	MIDI

Each byte can have any one of the following four values:

- 0 – Stop sending data
- 1 – Start sending data
- 2 – Stop receiving data
- 3 – Start receiving data

A control packet sent from the teachers program will either contain values for sending data or for receiving data. The sending and receiving codes cannot be combined in the same packet.

6.7 Performance

The performance of the system was measured in two areas. Firstly, the processor usage and the amount of data that was transmitted across the network and, secondly, the time difference from when the audio, video and MIDI data is captured to when it is played back. By comparing the time differences for the audio, video and MIDI the synchronisation of the multimedia data can be calculated.

6.7.1 Computer and Network Performance

The network utilization and CPU usage were measured while the system was being used. Measurements were taken while various combinations of audio and video were transmitted between the teacher and a student. The data was transmitted in one direction between the teacher and a student or in both directions. The Performance Monitor¹¹ was used to measure the bandwidth and CPU usage. MIDI messages were not included in the measurements since MIDI is limited to a maximum of 3125 bytes per second (about 1000 MIDI note-on messages), [Messick, 1998] which is relatively small compared to the number of bytes transmitted for audio and video. It is also highly unlikely that many MIDI messages will be generated per second during the music lesson for beginners. The different types of data being transmitted and direction of the transmission are labelled and are shown in table 11.

Table 11: Performance Monitor sections

<u>Section Label</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>
<u>Direction</u>	One direction	Both directions	One direction	One direction	Both directions	One direction
<u>Multimedia Data Transmitted</u>	Video	Video	Video	Video & Audio	Video & Audio	Video & Audio

Figure 27 illustrates the results of the performance test. Each marked section of the graph matches one of the labels from table 11. The graph illustrates the total number of bytes being transmitted across the network for the various combinations of audio and video transmissions. The y-axis has a scale of 10000 bytes.

Section A on the graph between the two marked vertical lines corresponds to the time that only video data was being transmitted in one direction. Each of the sections is approximately 15 seconds long with approximately a 2-second gap between each section.

¹¹ Performance Monitor – An administrative tool that is available in Microsoft Windows.

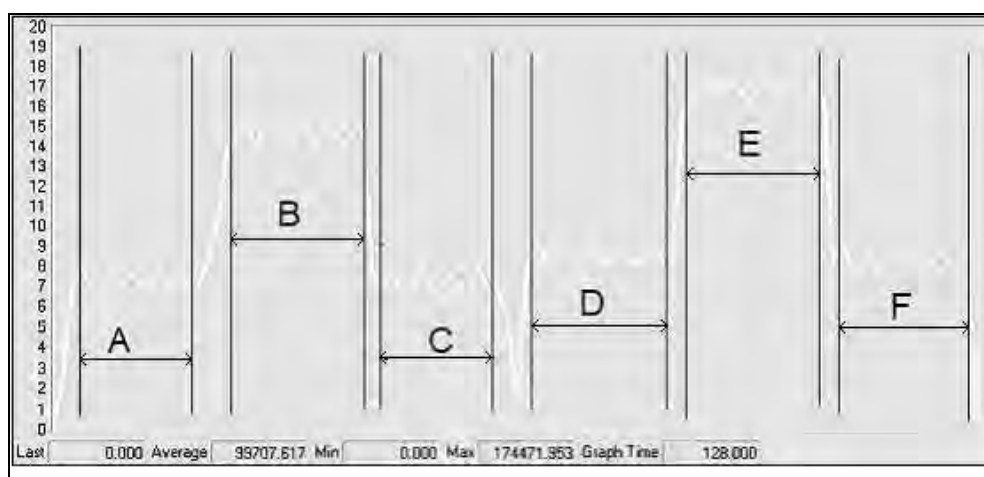


Figure 27: Performance Monitor graph

The amount of data being transmitted by the teacher is not affected by the number of students receiving the data since the data is broadcast across the network. The teacher is only able to receive data from a single student so it is not possible for more than one student to be transmitting multimedia data at any time. It is possible for the teacher to be transmitting to the whole class and only be receiving data from a single student. Again, the amount of data being transmitted from the teacher is the same as if he were transmitting to a single student.

The bytes per second measurements are shown in table 12 below.

Table 12: Bandwidth utilization

<u>Section</u>	<u>Bytes per second</u>
A	73276
B	145222
C	71323
D	85522
E	167618
F	83141

Table 13 shows the average CPU usage for the sections measured using the Performance Monitor. The CPU usage is measured as a percentage of the total CPU time available. As can be seen the results for the CPU usage for the teacher's computer and student's computer are similar.

Table 13: CPU usage

<u>Section</u>	<u>Teacher</u> <u>% CPU</u> <u>Usage</u>	<u>Student</u> <u>% CPU</u> <u>Usage</u>
A	15	16
B	23	25
C	16	16
D	16	16
E	24	26
F	16	17

6.7.2 Delays

It is important that an event such as a key being pressed on a music keyboard appears on the receiving computer as quickly as possible. If the delays are too long, then the communication between the teacher and student becomes unnatural.

The delay for the audio, video and MIDI were measured separately as described below.

6.7.2.1 Audio

Timings were recorded for the time differences between an event happening, and the result being heard by the receiver. The sounds were recorded using a wave file recorder in Windows. A single sharp sound was generated that was captured by the teachers program. The sound was transmitted across the network and played back on the student's computer through speakers attached. The volume was high enough so that the wave file recorder would record the original sound and the received sound.

A wave file editor was used to measure the time differences. The time from the start of the original sound to the start of the played back sound was measured. This is seen in the wave editor, in figure 28, as the highlighted section between the original sound and the received sound.



Figure 28: Sound delay

The first is the original sound, the second slightly softer is the sound played back from the speakers. The sounds following are echoes of the original sound. The results of these timings are shown in figure 29.

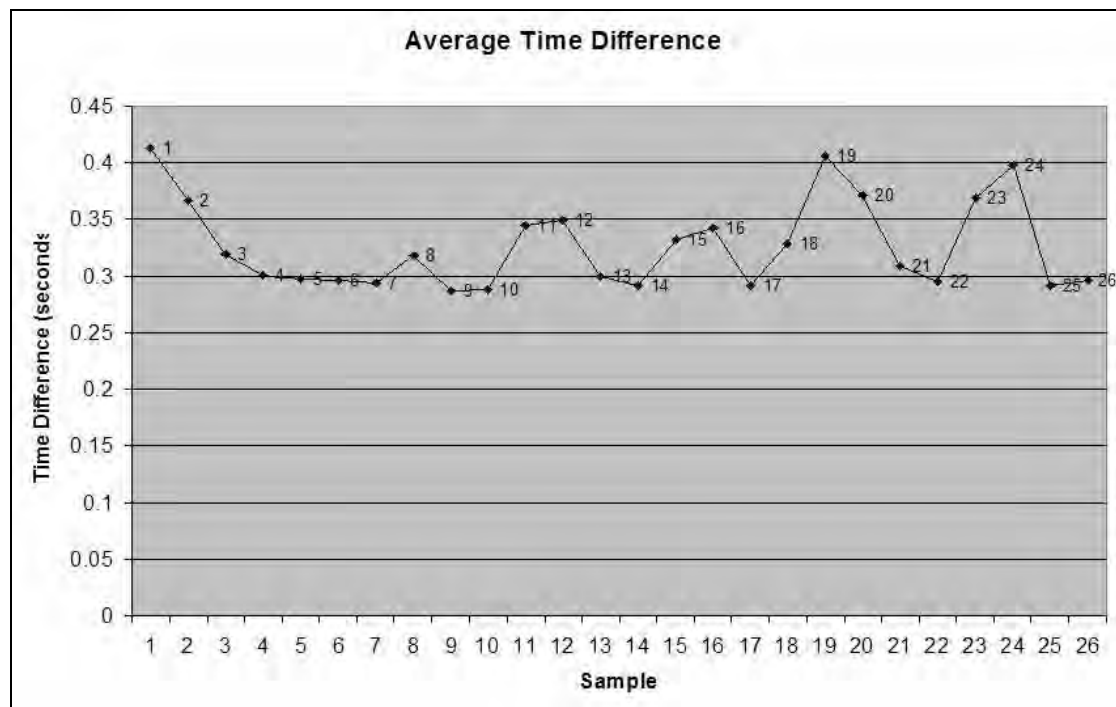


Figure 29: Audio delay

The average time taken for the sound to be captured, transmitted and played back is 0.32s. (The distance between the microphone, the sound source and the speakers was close, the estimate is that this only accounted for 2 or 3 ms delay)

6.7.2.2 Video

A video was taken using a video camera to measure the time delay between an event occurring, and the event being displayed on the remote view on the receiving program. Identical measurements were taken for the teachers program and for the students program as the receiving program.

The video camera was used to film the receiving screen and the original event at the same time. A hand was then moved and the results seen on the receiving computers screen. Using this video clip, the timings and delays were calculated.

The video clip was transferred from the video camera on the computer and saved as an AVI file. Using a video editor the clip was analysed and various measurements taken. Figure 30 shows a sequence of four frames of the video clip. The value under each frame is the time stamp for the frame.



Figure 30: A video clip for the student's program

Table 14 shows the time delay between the original event occurring and the event being seen on the receiving video view for the student's program. Table 15 shows the results for the teacher's program. The average delay was 0.17 seconds for both tests.

Table 14: Results for the student's program

Start Frame Number	End Frame Number	Frame Difference	Start Frame Time	End Frame Time	Time Difference
646	649	3	43.066	43.266	0.2
680	682	2	45.332	45.466	0.134
714	717	3	47.599	47.799	0.2
747	749	2	49.799	49.932	0.133
761	764	3	50.732	50.932	0.2
1037	1040	3	69.132	69.332	0.2
1061	1064	3	70.732	70.932	0.2
1094	1097	3	72.932	73.132	0.2

Table 15: Results for the teacher's program

Start Frame Number	End Frame Number	Frame Difference	Start Frame Time	End Frame Time	Time Difference
167	170	3	11.133	11.333	0.2
195	198	3	12.999	13.199	0.2
222	224	2	14.799	14.933	0.134
247	250	3	16.466	16.666	0.2
274	276	2	18.266	18.399	0.133
326	328	2	21.733	21.866	0.133
354	356	2	23.599	23.733	0.134
370	372	2	24.666	24.799	0.133

6.7.2.3 MIDI

Timings for the MIDI were approached slightly differently. It is possible for any MIDI messages sent to the student from the teacher to be sent back to the teacher. A recording of the MIDI notes being played on the teacher's keyboard and the subsequent note that is played back by the sound card was made. The time difference between the first note and the subsequent note gives the time taken for the note to be transmitted to the student and back to the teacher. Table 16 gives the results of these timings in seconds.

Table 16: MIDI delay for the teacher

<u>No</u>	<u>Time difference</u> <u>(seconds)</u>		<u>No</u>	<u>Time difference</u> <u>(seconds)</u>
1	0.038		6	0.033
2	0.047		7	0.021
3	0.046		8	0.032
4	0.028		9	0.025
5	0.049		10	0.040

The student's MIDI configuration was modified so that similar timings could be made from the student's program. The results were very similar to the first set as seen in table 17.

Table 17: MIDI delay for the student

<u>No</u>	<u>Time difference</u> <u>(seconds)</u>		<u>No</u>	<u>Time difference</u> <u>(seconds)</u>
1	0.034		6	0.045
2	0.041		7	0.037
3	0.046		8	0.041
4	0.028		9	0.032
5	0.044		10	0.048

From the two sets of results, it can be seen that the delay between a MIDI message being generated to when it is heard on the receiving end is very small.

6.7.3 Synchronisation

The synchronisation of the multimedia data has not been dealt with explicitly in the implementation. The audio, video and MIDI do appear to be sufficiently synchronised for the system to be used without difficulty. By varying the number of buffers used for the audio and video data, it was possible to have the multimedia appear synchronised. (The number of buffers used affected the delay of the multimedia data.)

Using the results from the previous section (6.7.2.1 and 6.7.2.2) the synchronisation of the multimedia data can be examined.

Audio delay = 0.32 seconds

Video delay = 0.17 seconds

It takes on average 0.32 seconds for the audio data to be played back by the receiving computer and an average of 0.17 seconds for the receiving computer to display the video images. From that, the average time difference between the audio data being played and the video data being displayed is 0.15 seconds. (i.e. $0.32 - 0.17$)

Although the time difference between the audio and video is large for this system, the delays are consistent and the system can be used without difficulty.

6.8 Summary

The implementation of the remote interactive keyboard teaching system is based on the design developed in the previous chapter. Since most of the reviewed software runs on the Microsoft Windows operating system and it is widely used, the system was developed to run on the same operating system.

The hardware requirements for the system were discussed along with the necessary developers kits needed to access the hardware devices. The code was written using Visual C++ that supports MFC and the required developer kits.

The implementation of the GUI was guided by the requirements specified in the design. The GUI needed to be simple to use and only offer the necessary features that would be used in the class.

UDP was used as the network protocol. This is a fast and reliable protocol on a LAN. The packet structures were defined for controlling what multimedia data to send and who should be receiving the data. The protocol for managing the students joining and leaving the class was described and which ports were used for the different types of packets.

The audio data was captured and sent using the default audio settings, PCM. The audio was captured using the Video for Windows libraries. The same libraries were used for the capture of the video images. The video images were in the bitmap format.

A possible configuration for MIDI was described. How the system is configured to use MIDI can be changed depending on what is needed for the class. Unlike the audio and video, there can be more than one source of MIDI messages in the system. A MIDI patch program was used to route MIDI to multiple devices.

Application sharing was available through VNC. The teacher can select any student to see and control that student's computer using VNC. It is started from the teacher's program.

Once the system had been implemented, various measurements were taken to measure its performance. The number of bytes transmitted was measured for different combinations of audio and video being transmitted between the teacher and a student while monitoring the CPU usage.

Delays and synchronisation for the different types of multimedia were examined. The times taken for a sound to be played back on the receiving end were measured. Similar measurements were taken for the video images. The results showed that although the two data streams were not synchronised with each other, they were close enough for the system to be usable in order to demonstrate how such a teaching system would perform.

6.9 Concluding Remarks

Although the application sharing and MIDI routing were handled by third party applications, WinVNC and MIDIYoke, it is possible to implement these abilities into the system. This would present the teacher with a single consistent interface instead of relying on three different applications and styles of interface. It was decided to use the third party solutions since these were readily available. The addition of these capabilities in the code would have required a significant amount of effort without affecting the performance of the system.

The implementation of the design did not incorporate video compression. The quality of the video images chosen for the implementation was high enough for the system to be used and tested. Adding the compression would have made the implementation more complex with the only benefit being a clearer video image been transmitted and displayed. If video

compression were implemented a larger video image could be used and at a higher frame rate without having to transmit any more bytes across the network.

As the processors are becoming faster and the compression algorithms improving it is possible to compress a live video stream.

7 Conclusion

The principles of learning how to play an instrument have not changed markedly over the years. New theories about the most effective way to teach music skills as well as how a student learns to play an instrument have emerged, but one of the vital aspects of a music lesson is still the interaction between student and teacher. As new computer-related technology becomes more widely available and affordable, computers are beginning to play a larger role in music education.

Learning how to play the keyboard is a very practical undertaking and during a music lesson, a lot of interaction is required between the teacher and the student. The music lesson is teacher intensive. The teacher not only demonstrates techniques used, but also watches and corrects a student's technique. Due to the nature of music lessons a teacher normally only has one person at a time for a lesson. This can be expensive and impractical for many music students. The ideal scenario is for a teacher to make use of modern technology to teach music skills to a number of students at once.

A selection of available music software programs have been evaluated in this thesis for their use as tools in learning how to play the keyboard. As sophisticated as the available software programs are for learning how to play the electronic keyboard, they do not offer the student the all-important human factor, that is, the interaction, motivation and correction of a music teacher. Many software programs are intended for use by an individual and not in a music class. Specialised education systems were considered but these systems are not designed for use in a music class as there is no support for MIDI transmissions. The use of video conferencing applications do allow the teacher and student to interact however, they are not suited for the classroom structure and do not support any MIDI transmissions.

In this thesis, an ideal system has been described that allows a music teacher to teach a class keyboard skills with the same level of interaction found in a one on one tutored lesson using up to date computer technology. The system allows for audio, video and MIDI to be transmitted over a network to enable the teacher to interact with individual students in a music class during a lesson. The teacher is able to monitor individual students and can see and hear what the student is doing. MIDI messages are used to transmit the musical information between the teacher and student. These MIDI messages can be sent to the music programs running on the student's computers.

A solution was designed based on the description of an ideal system. The design was developed using the UML notation and Rational Rose. The analysis and design steps described by Unified Process were used for the design. The design consists of two separate programs, one for the teacher and another program for the student. The design was implemented using Microsoft's Visual C++ and the Microsoft Foundation Classes. The Video for Windows Developer Kit and the classes provided by the Maximum MIDI Toolkit were used for the capture and playback of the audio, video and MIDI data.

One of the key issues for the performance of the system was to do with ease of use. The system had to be used as a tool in a music lesson without distracting the teacher and student from the lesson itself. This issue was dealt with by providing a simple interface for the system that meant that all the technical details were automatically pre-configured in the code of the programs used by both teacher and student.

The performance of the system was measured. The CPU usage and the bytes transmitted were noted. The tests were carried out while different combinations of audio and video were being transmitted between a teacher and a student. The results of the measurements showed that although there was a delay in the multimedia transmission the delay was not erratic and the system was usable. The teacher and student would not be distracted by using the system.

The aim of this thesis was to show that distributed multimedia could be adapted and used for teaching keyboard skills in a classroom. This thesis has addressed the issues involved in teaching and learning keyboard skills. It has looked at the shortfalls of software available for use by music teachers and students alike. It has explored the possibilities of computer assisted classroom music tuition. An ideal solution has been designed and implemented to address the need for both teacher input and the use of computer software packages in music tuition. The effectiveness of the solution has been discussed and various performance issues have been noted.

A solution was designed and implemented. The implementation of the design proved that it is possible to use a computer as a tool for remote interactive keyboard lessons through the successful transmission of audio, video and MIDI data.

Bibliography

- Alfred Alfred Publishing, "*Essentials of Music Theory*", <http://www.alfred.com>
- Alvear, 1998 Alvear, J. "*Web Developer.com Guide to Streaming Multimedia*", John Wiley & Sons, 1998
- Britannica "*Encyclopaedia Britannica*", <http://www.britannica.com>
- Chancery Chancery, "*WinSchool*", <http://www.chancery.com>
- CrossTec CrossTec, "*NetOpSchool*", <http://www.crossteccorp.com>
- Fowler, 2003 Fowler, M. "*UML Distilled*", Addison Wesley, 2003
- Goddard, 1999 Goddard, R "*Technology in Music Teacher Education*", <http://music.utsa.edu>, 1999
- Greunen, 2000 Greunen, D., Wesson, J. L. "*Synchronous Learning: Using IT to bridge the gap in Distance Education*", Conference on Information Technology in Tertiary Education, June 2000
- Heath, 1999 Heath, S. "*Multimedia Communications Technology*", Focal Press, 1999
- Independent Education, 2002 Independent Education, "*Schools will teach maths by video link to overcome staff shortages*", <http://education.independent.co.uk>, 21 May 2002
- Jacobson, 1999 Jacobson, I., Booch, G., RumBaugh, J. "*The Unified Software Development Process*", Addison-Wesley, 1999
- Jump Music Jump Music, "*Piano Discovery*", No longer exists although product still available
- Kirk, 1999 Kirk, R., Hunt, A. "*Digital sound Processing for Music and Multimedia*", Focal Press, 1999
- Larman, 2001 Larman, C. "*Applying UML and Patterns*", Prentice Hall, 2001
- MakeMusic MakeMusic Inc, <http://www.finalemusic.com>
- McConnell, 1993 McConnell, S. C. "*Code Complete: A Practical Handbook of Software Construction*", Microsoft Press, 1993
- McKay, 1999 McKay, E. N. "*Developing User Interfaces for Microsoft Windows*", Microsoft Press, 1999
- Messick, 1998 Messick, P. "*Maximum MIDI*", Manning Publications, 1998
- MiBAC MiBAC Music Software, <http://www.mibac.com>

Microsoft	Microsoft, http://www.microsoft.com
Microsoft Press, 1997	Microsoft Press, " <i>Networking Essentials</i> ", Microsoft Press, 1997
MidiTec	MidiTec, " <i>Earmaster</i> ", http://www.earmaster.com
MIDIYoke	MIDI-OX, " <i>MIDI Yoke Junction</i> ", http://www.midiox.com
Midwest, 2003	Midwest Clinic, " <i>An International Band and Orchestra Clinic</i> ", http://www.midwestclinic.com , August 2003
Miller, 1998	Miller, K. C. " <i>Multicast Networking and Applications</i> ", Addison-Wesley, 1998
Mueller, 1998	Mueller, S. " <i>Upgrading and Repairing PCs</i> ", Que Corporation, 1998
Musicware	Musicware, No longer exists, product still available
NAMM	NAMM, " <i>International Music Products Association</i> ", http://www.namm.com
OMG	Object Management Group, " <i>Unified Modeling Language</i> ", http://www.omg.org
Palmer, 1992	Palmer, K. " <i>Teach Yourself Music</i> ", Hodder & Stoughton, 1992
Papanikolaou, 2000	Papanikolaou, G, Kalliris, G. K., Dimoulas, C., Pasiadis, C., Galatsopoulou, F. "Deployment of Internet Resources for Digital Audio Learning Course", Proceedings of the AES 108th Convention, Paris, 2000, 2000
Penton, 2000	Penton, J " <i>Deploying an H.323 Environment to Provide a Desktop Multimedia Conferencing Environment to the CS department</i> ", Honours project report, Department of Computer Science, Rhodes University, November 2000
PGMusic	PG Music Inc, http://www.pgmusic.com
Pressman, 2000	Pressman, R. S. " <i>Software Engineering - A practitioners Approach</i> ", McGraw-Hill Publishing Company, 2000
Prosis, 1999	Prosis, J. " <i>Programming Windows with MFC</i> ", Microsoft Press, 1999
Quatrani, 1999	Quatrani, T. " <i>Visual Modeling with Rational Rose and UML</i> ", Addison Wesley, 1999
Quatrani, 2001	Quatrani, T. " <i>Introduction to the Unified Modeling Language</i> ", Rational Unified Conference, 2001
RealVNC	RealVNC, http://www.realvnc.com
Rising Software	Rising Software, http://www.risingsoftware.com
Roland	Roland, http://www.roland.com

- Rothstein, 1992 Rothstein, J. "*MIDI A Comprehensive Introduction*", Oxford University Press, 1992
- Sibelius The Sibelius Group, "*Sibelius 2*", <http://www.sibelius.com>
- Spruce, 1996 Spruce, G. "*Teaching Music*", Routledge, 1996
- Statmarket Statmarket, <http://www.statmarket.com>
- Steinberg Steinberg, "*CuBase VST*", <http://www.steinberg.net>
- Symantec Symantec, "*PCAnywhere*", <http://www.symantec.com>
- Taylor, 1990 Taylor, E "*Music Theory in Practice*", Associated Board of the Royal School of Music, 1990
- Twelve Tone Systems Twelve Tone Systems, "*Cakewalk*", <http://www.cakewalk.com>
- Voyetra Voyetra Music Software, *Voyetra Turtle Beach Inc*, <http://www.voyetra.com>
- Zeigler Zeigler, "*Choosing and using Music Software in the Studio and Home*", <http://www.pianoeducation.org>

Appendices

A) Flow of Events: Teacher and Student

Teacher: Login the teacher

The teacher starts his program.

The program starts and the audio and video preferences are set for the teacher's audio and video devices.

A window is opened showing the local camera's view, the remote video camera's view, the transmission list, the receiving list and the assistance list.

Student: Leave the class

The student clicks on the 'Logout' button.

The logout request is sent across the network to the teachers program.

The student's program stops transmitting any multimedia data that is being transmitted and stops processing any multimedia data it is receiving.

The student's window is closed and the students program ends.

Teacher: Logout the student

A logout request is received from the student program from the network card.

The request is passed to the teachers program.

Any multimedia data that is being sent to the student is stopped and any multimedia data received from the student is not processed.

The students name is removed from the transmission list, receiving list. The students name is removed from the assistance list if the student had asked for assistance.

The student's name and IP address are removed from the class list.

Teacher: End the class

The teacher clicks on the 'End Class' button on the teachers window.

The teacher's program receives the request and stops transmission and receiving of multimedia data.

An end of class message is sent to each of the students that are part of the class across the network through the network card.

All the students in the class are removed from the class list.

The teacher's window is closed and the program ends.

Student: End of class

The 'End Class' message is received from the network.

The students program receives the message from the network card and closes all multimedia communication.

The student's window is closed and the program ends.

Teacher: Receiving multimedia data

The teacher selects the student who will start sending multimedia data to the teacher.

The teachers program stores the name of the student.

The type of multimedia data the teacher expects to receive from the student is then selected.

The teacher clicks the 'Receiving' button.

The teachers program sends the 'Start Transmission' request to the selected student with the types of multimedia the student program is to send.

The message is sent to the network card to be transmitted across the network.

The teacher clicks on the 'Stop Receiving' button.

The message is sent to the teachers program.

A message is sent to the student to stop transmitting multimedia data to the teacher through the network card.

The teachers program stops processing multimedia data it receives.

Student: Transmit multimedia data

A message is received from the network indicating the student is to transmit multimedia data.

The packet is passed to the student's program.

The packet is checked to see what type of multimedia data is to be sent to the teacher.

The appropriate hardware device then captures the selected types of multimedia data.

The captured data is sent to the network card for transmission across the network to the teachers program.

Student: Close multimedia communication

The stop transmission packet arrives at the network card.

The packet is passed to the students program.

The students program stops capturing the specified multimedia data.

Student receives a stop receiving multimedia data request.

The network card receives a stop receiving packet.

The packet is passed to the student's program.

The student's program stops processing the multimedia data it receives.

B) Sequence Diagrams

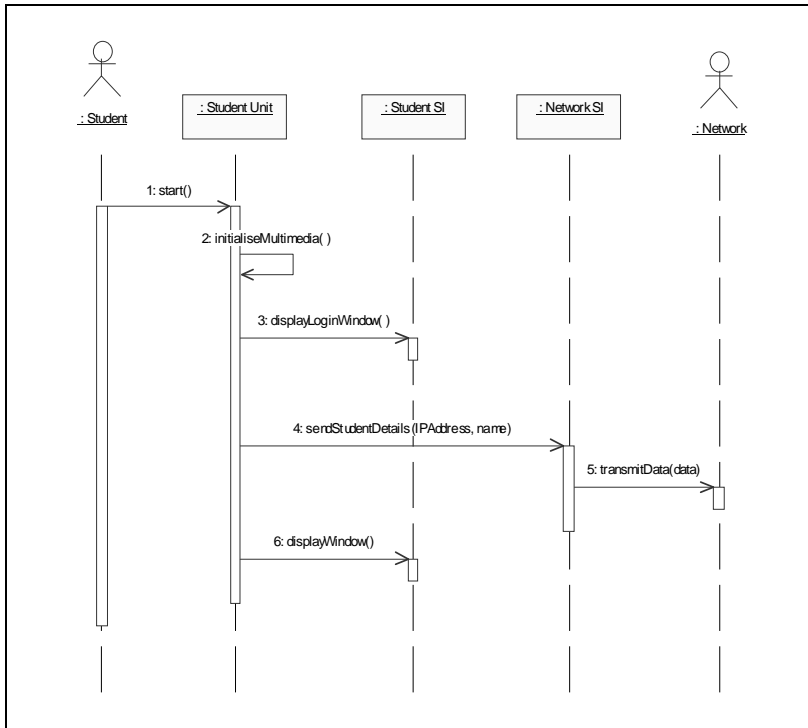


Figure 31: Student- Join the class

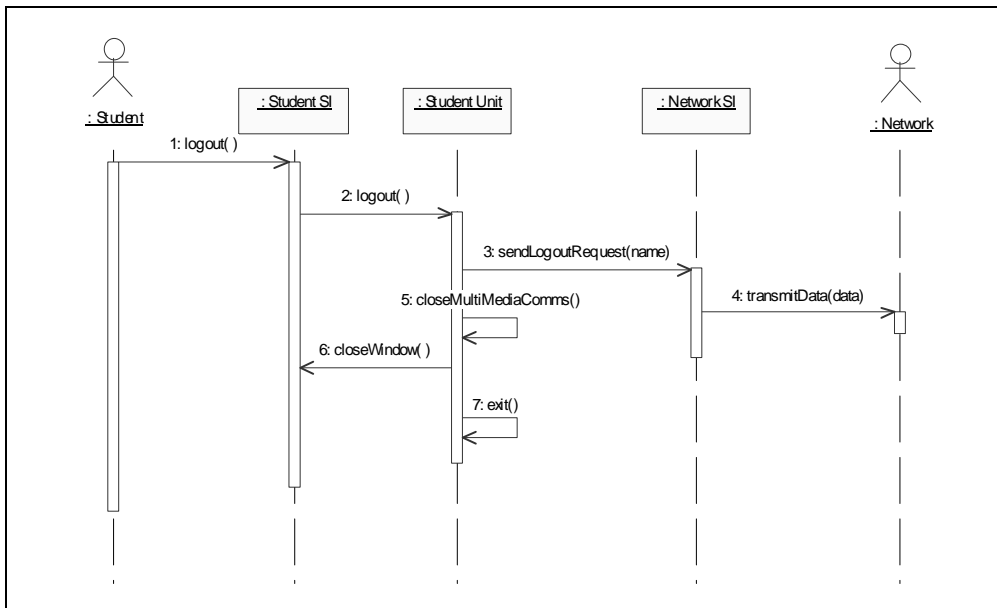


Figure 32: Student- Leave the class

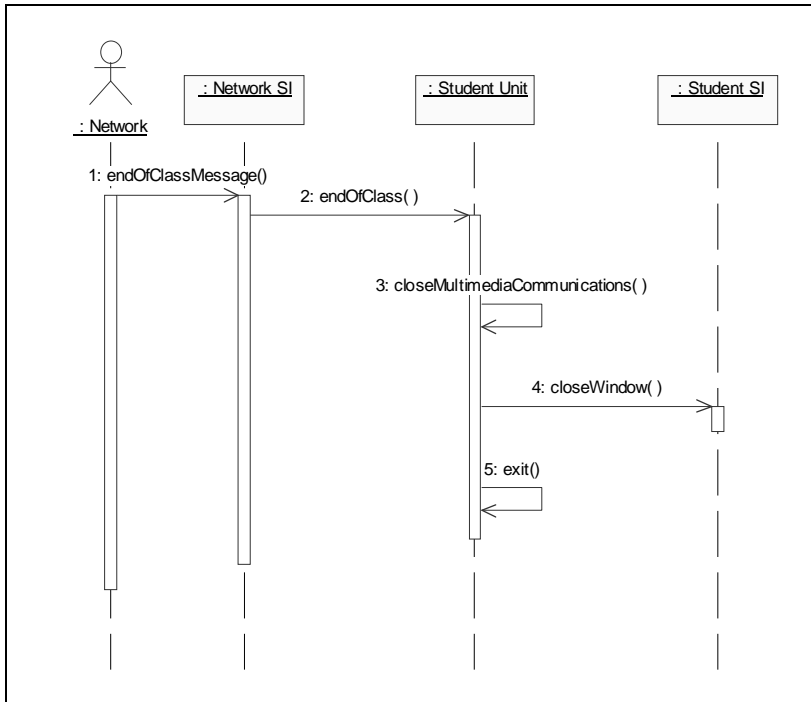


Figure 33: Student- End of class

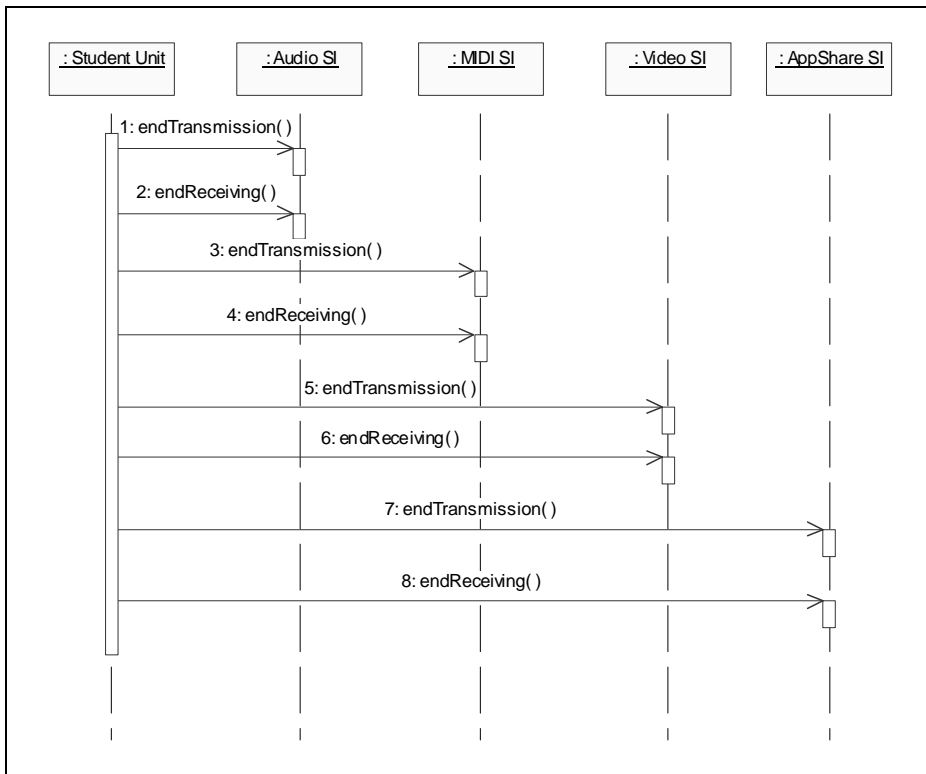


Figure 34: Student- Close multimedia communications

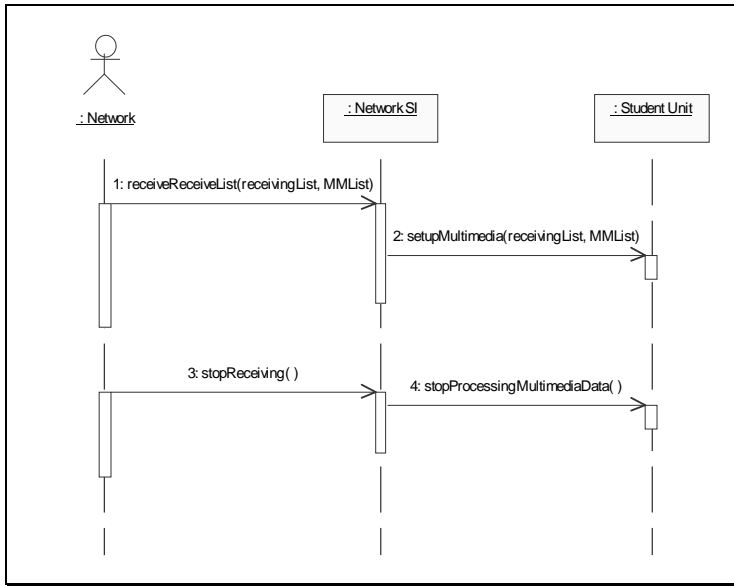


Figure 35: Student- Receives receiving request

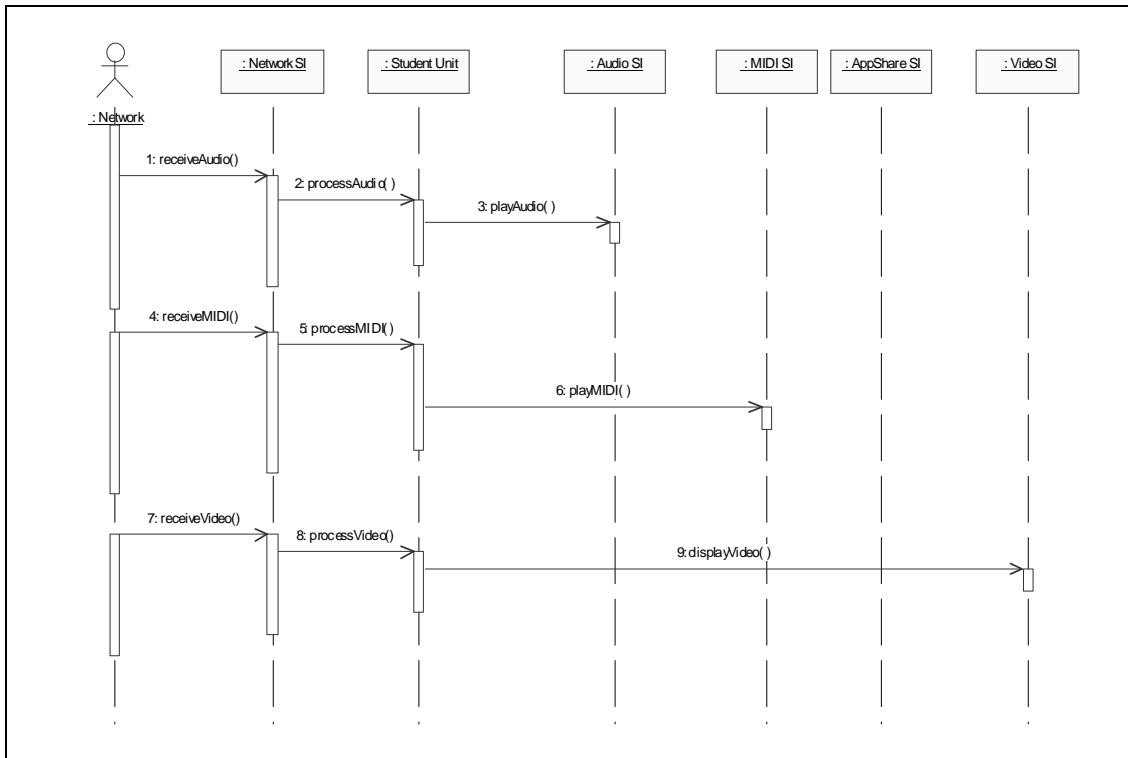


Figure 36: Student- Receives multimedia data

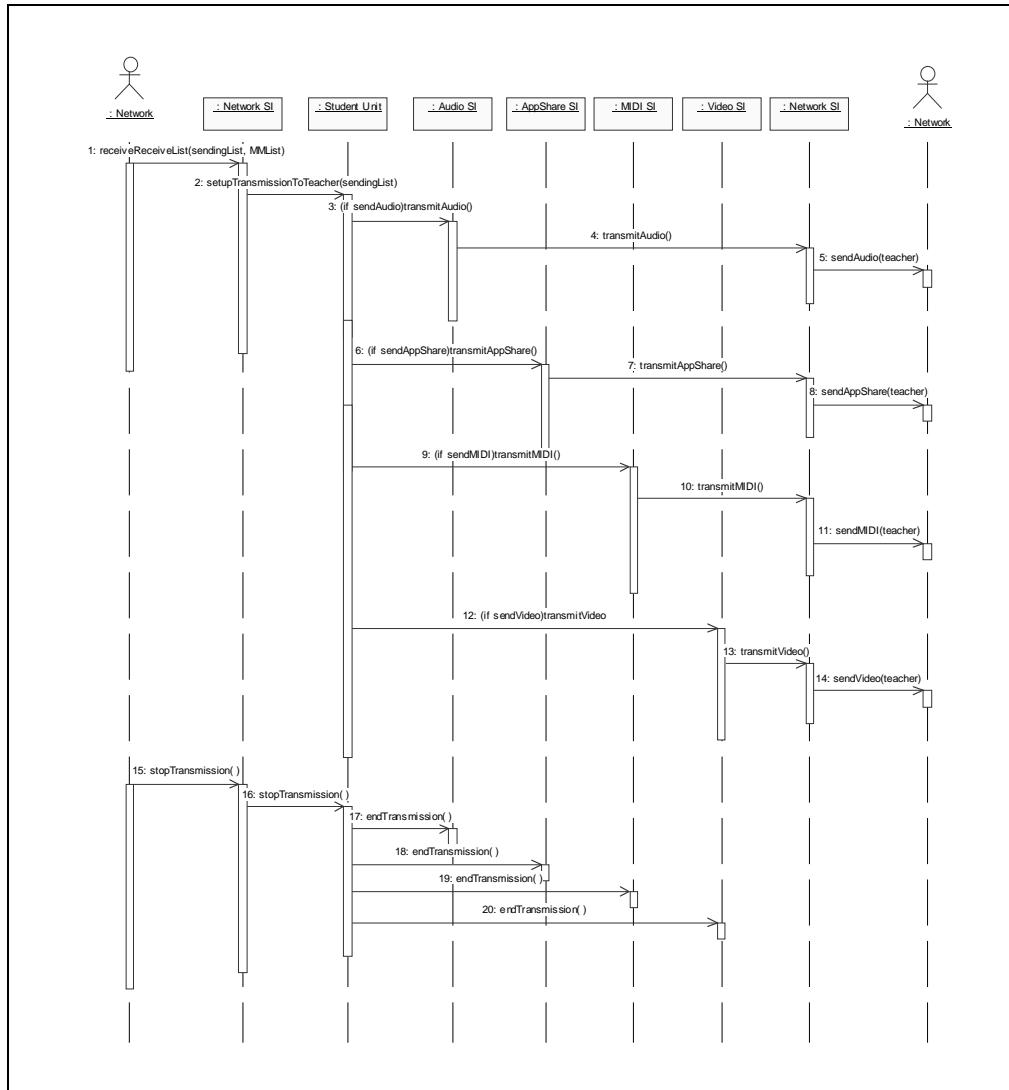


Figure 37: Student- Transmit multimedia data

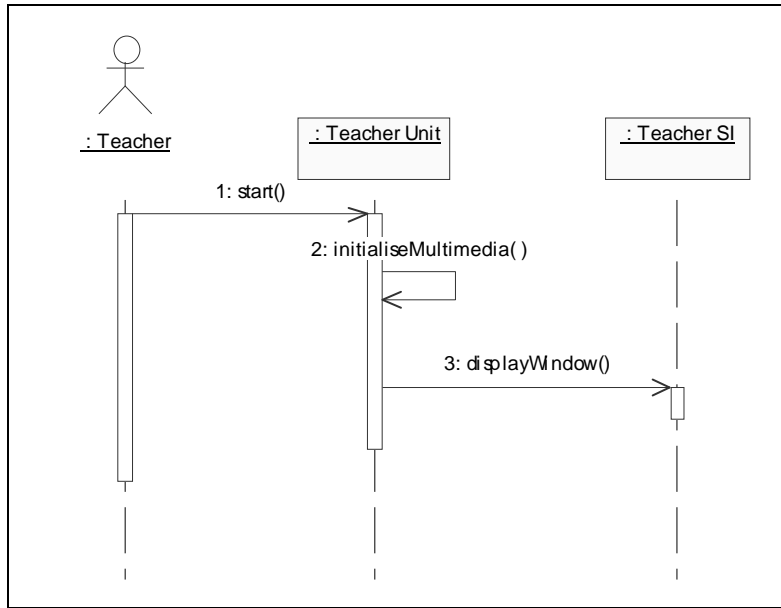


Figure 38: Teacher- Start the class

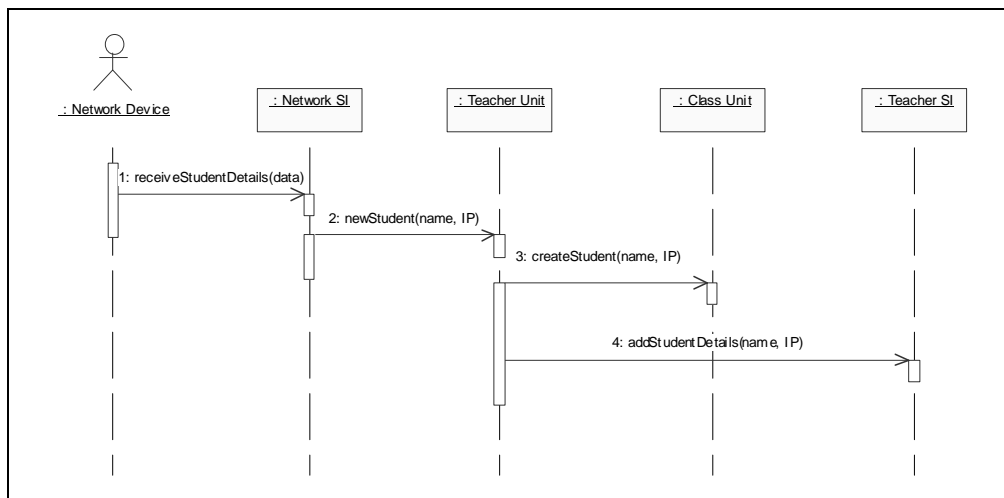


Figure 39: Teacher- Add a student

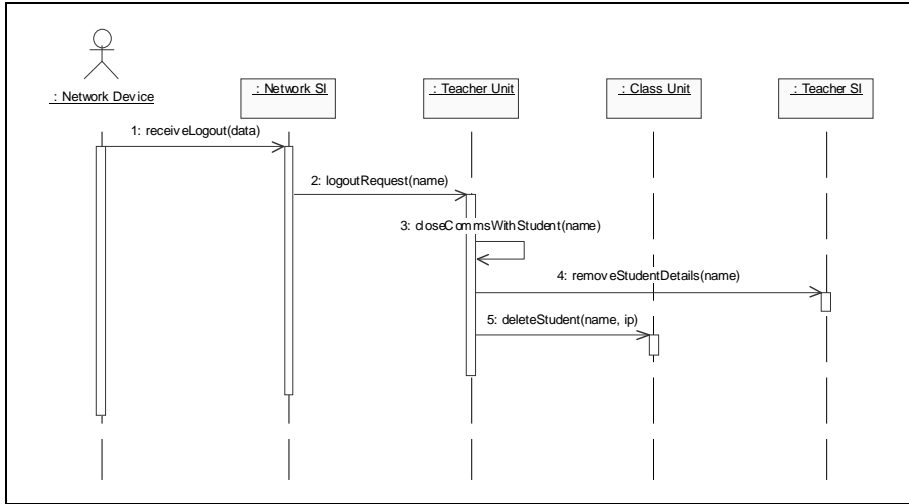


Figure 40: Teacher- Remove a student

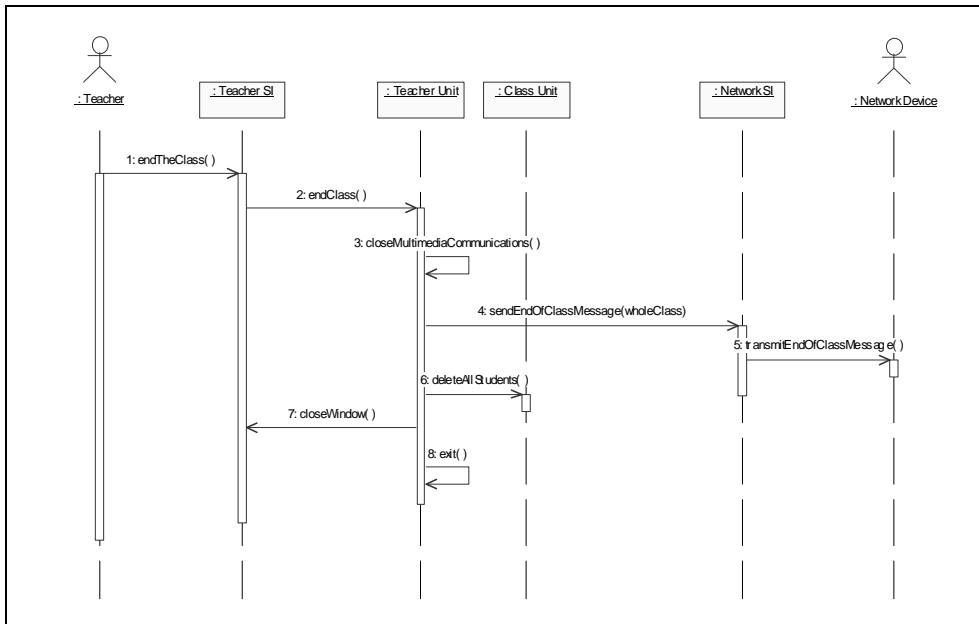


Figure 41: Teacher- End of class

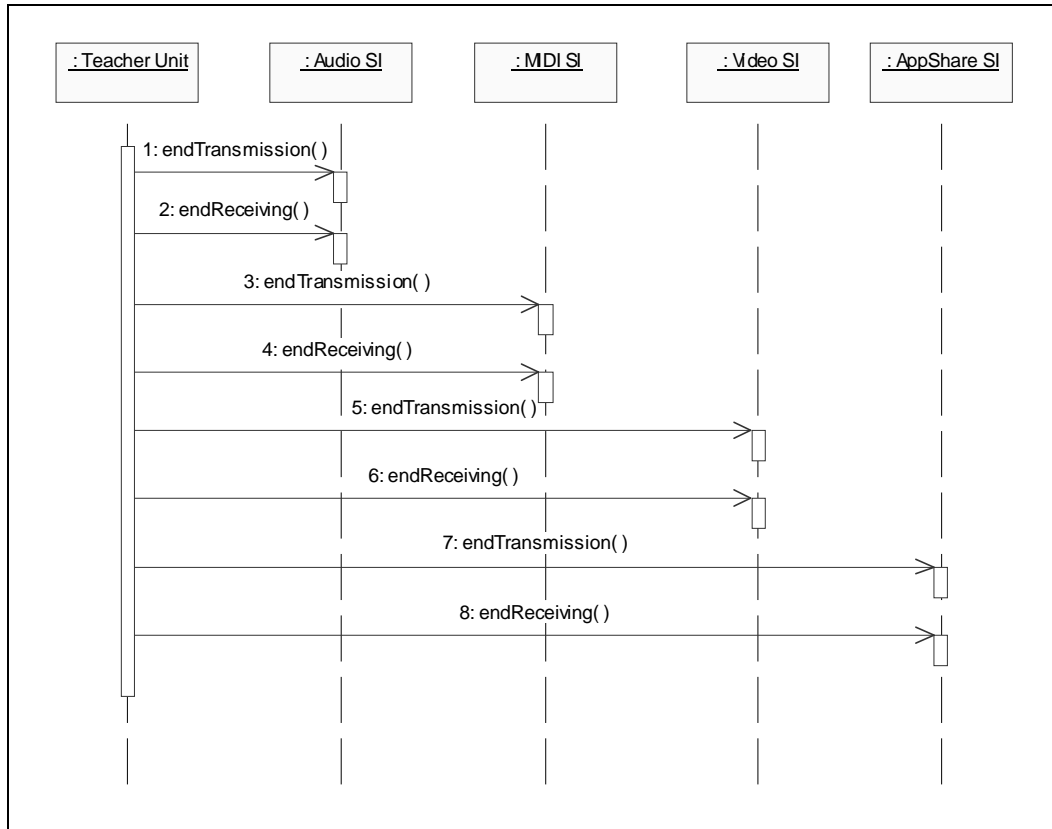


Figure 42: Teacher- Closes multimedia communications

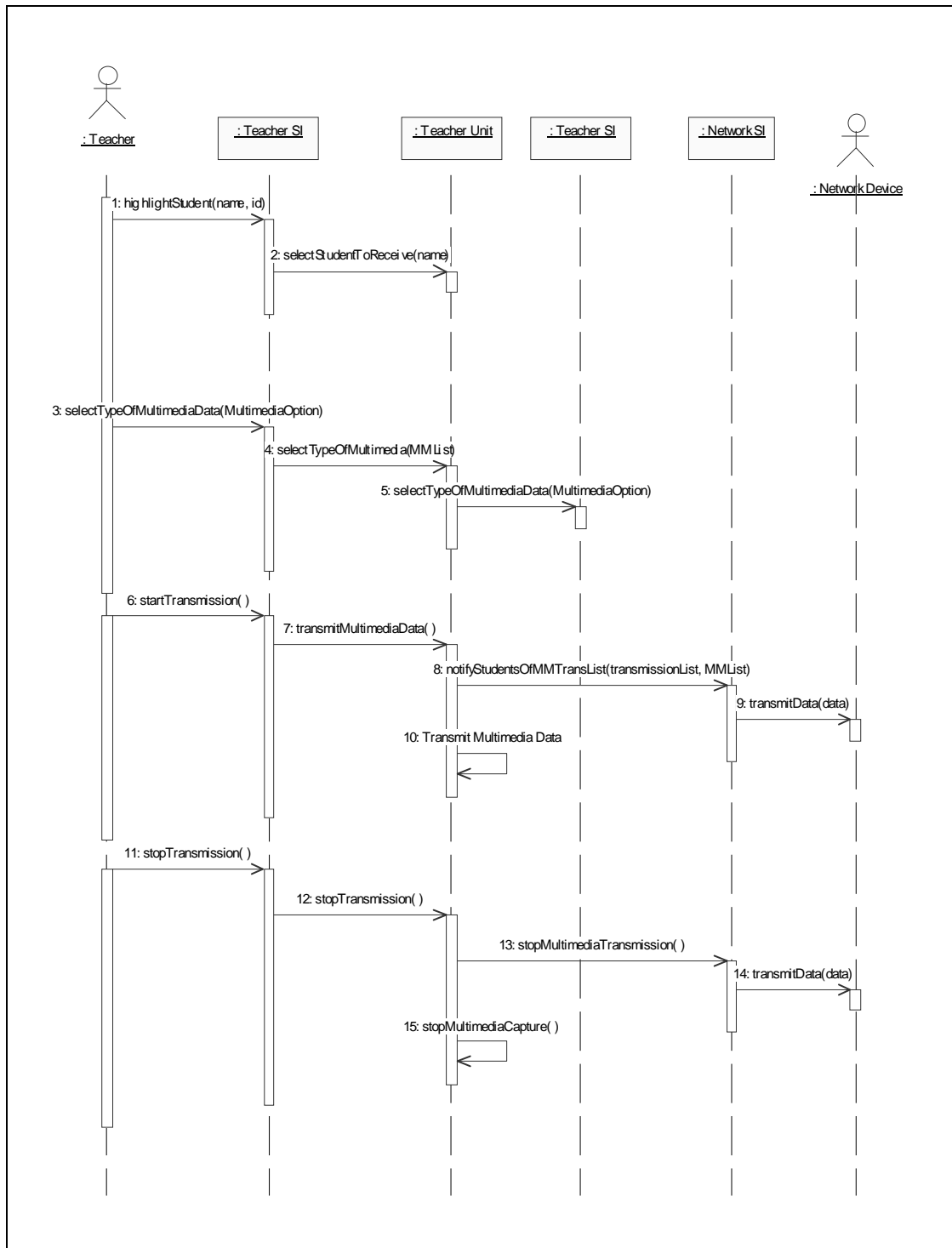


Figure 43: Teacher- Start/stop multimedia transmission

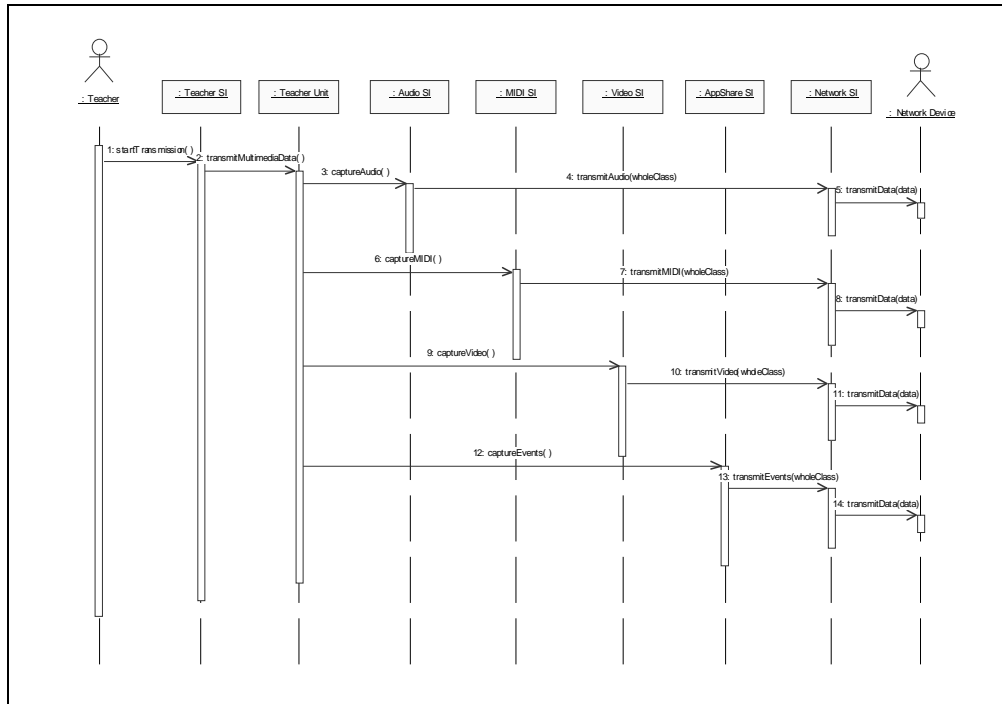


Figure 44: Teacher- Transmit multimedia data

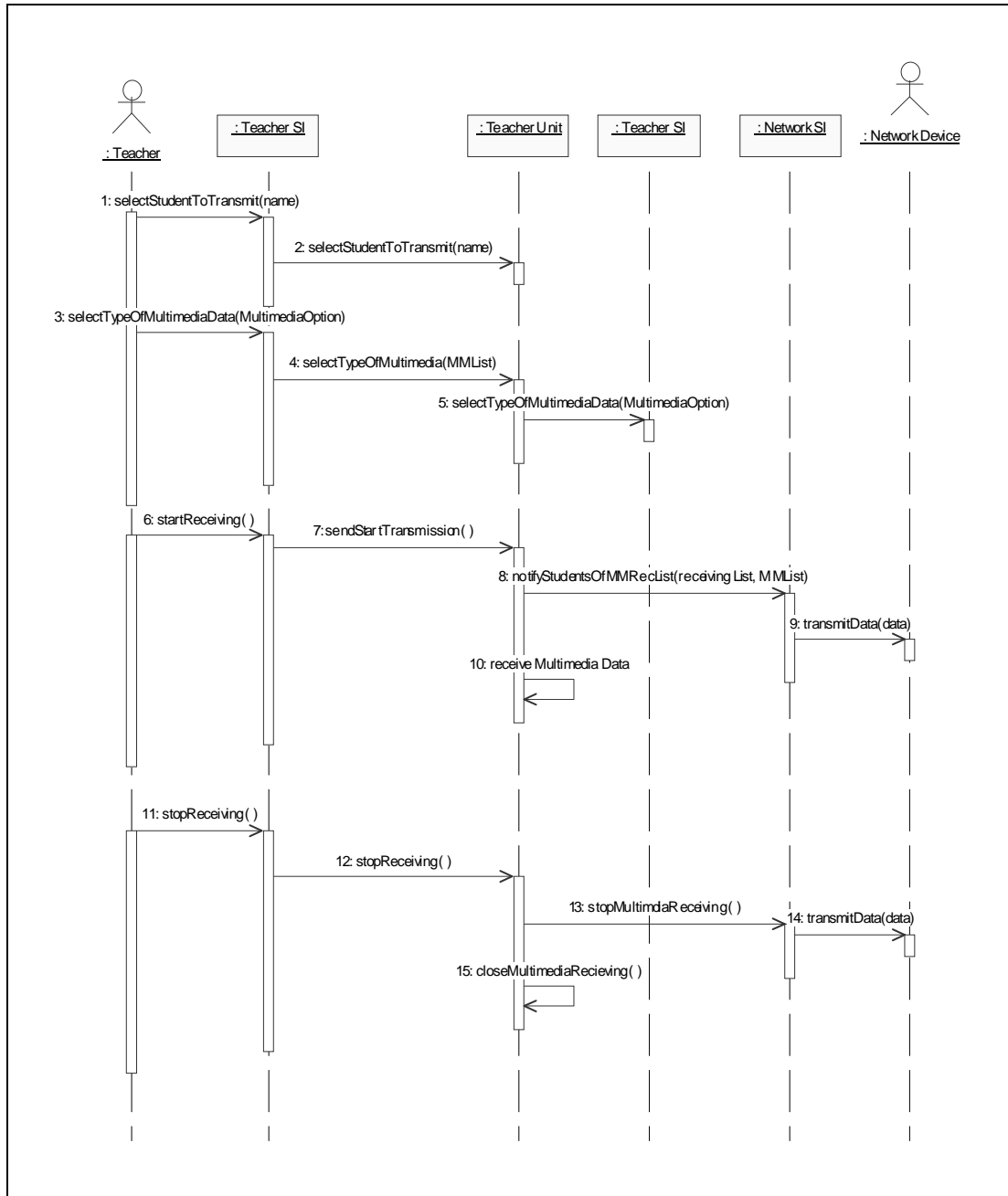


Figure 45: Teacher- Starts/stops multimedia receiving

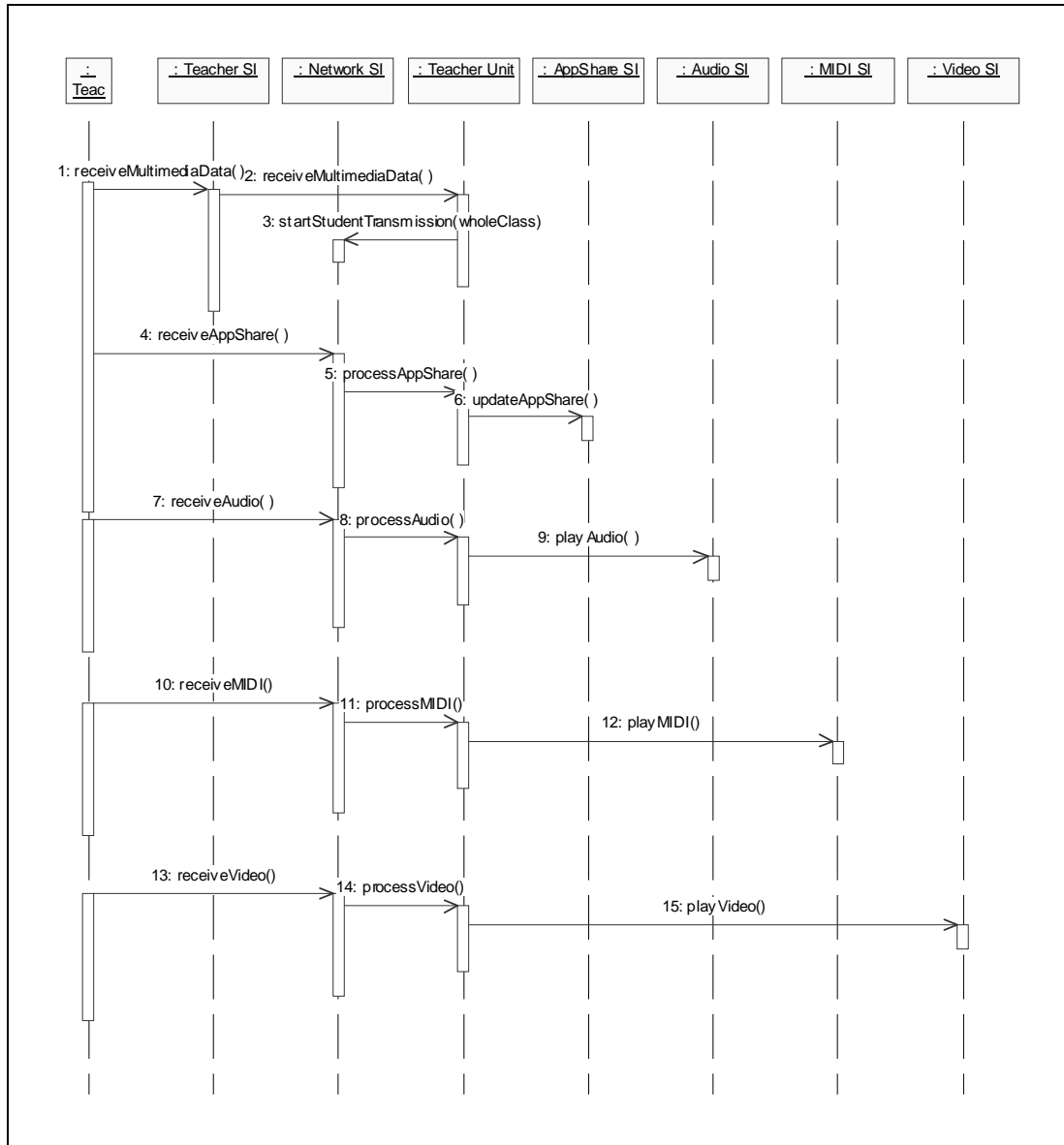


Figure 46: Teacher- Receives multimedia data