Encoding 3D sound scenes and music in XML

Guillaume Potard[1], Stephen Ingham[2]

[1]School of Electrical and Telecommunication Engineering, [2]Faculty of Creative Arts,
University of Wollongong
email: gp03@uow.edu.au, singham@uow.edu.au

Abstract

This paper presents an ongoing research project taking place at the University of Wollongong which aims to develop a hardware and software framework for the creation, manipulation and rendering of complex 3D sound environments described in XML format. The proposed system provides the composer with a platform where virtual objects such as sound sources, reflective surfaces, propagating mediums and others can be used artistically to create time varying virtual scenes. The Extended Markup Language (XML) is used to describe and save the content and temporal behaviour of virtual sound scenes or musical compositions. The XML encoded scenes are then parsed by a Java application which in turn sends real-time commands to a signal processing layer implemented in MAX/MSP. Ambisonics 4th order on a 16-speaker dome is used for spatialisation.

1 Spatial music: a background

Spatial location as an explicit parameter of musical composition has existed at least since the work of the Gabrielis and the *cori spezzati* of sixteenth-century Venice. However, it was not until as recently as the post-WWII era that composers such as Varese (*Poeme Electronique*, 1958), Xenakis (*Bohor 1*, 1962), Stockhausen (*Spiral*, 1970) and Chowning (*Turenas,* 1972) conceived a far more profound synthesis of sound and architectural design. Experimental sound projection spaces such as Le Corbusier's Philips Pavilion (Brussels World Fair, 1958) and the Stockhausen/Bornemann spherical auditorium in Osaka (Expo, 1970) were in some respects the precursors of today's cinematic sound diffusion protocols.

However, these spaces were constructed for the performance of specific artworks rather than for any more general virtual sound scene creation. Moreover, the uniqueness and site-specificity of their design ruled out any possibility of wider applications.

Since this time, research centres such as IRCAM (Paris) and the Music Technology Group (University of York) have pursued both the theory and practical aspects of sound localisation. The Berio/di Giugno TRAILS project (Florence, 1979) was typical of the new approach to multi-speaker networking which culminated in the development of programming environments such as MAX (Matthews, Puckette *et al*.) These, in turn, have provided the tools for projects such as the one presently under discussion.

Today's technologies offer exciting perspectives for musical composition in three dimensions. For instance, when sound sources have distinct spatial locations, the binaural system is able to listen and separate a large number of simultaneous auditory streams; this ability is known as the Cocktail Party Effect (Arons, 1992). As a result, very complex soundscapes made of many layers of sounds can be devised, while the listener remains able to maintain a focus on individual parts of the scene. Three-dimensionality also offers true immersion in a virtual sound environment. This perception of total envelopment is almost always missing with stereo techniques.

2 Design objectives

The main aim of this project is to provide composers with an intuitive system by which virtual 3D sound scenes can be created and played to an audience. The scenes are composed of virtual objects that are the translation of real sound objects such as sound sources into a virtual acoustical space. Being object oriented, the proposed method is easy to assimilate and scenes easy to interact with. In addition, a hierarchy between objects can be set in the virtual sound scenes. This hierarchy allows the creation of complex sound objects composed of several elementary objects such as sound sources and acoustic surfaces. Once defined, these complex objects can be saved separately in XML format and be re-imported in further scenes.

To describe the temporal behaviour of sound scenes, a concise scene score is used so that the composer is able to set time events and parameter changes in the scenes.

One other important aim of this project is to provide the composer with a 3D visual interface

implemented in Java3D. This visual interface facilitates visualisation, interaction and creation of the virtual sound scenes.

The XML mark-up language is used as a saving and exchange format as it is clearly set out, easily readable and works well with the object-oriented approach of our virtual scene description scheme.

The proposed system can be used to create virtual sound scenes and play them back at a later stage. During playback, object positions and parameters can also be modified in real-time, leaving some freedom to the performer during playback. Also live microphone inputs and network streams can be used instead of recorded sound samples so that live instruments can be incorporated in the virtual sound scene. The system can also directly import B-Format recordings (Gerzon 1985), (Soundfield) so that hybrid 3D audio scenes that are both composed by recorded real sound scene and spatialised monaural sounds can be constructed.

An overview of the system is given in Part 2. The processing of scene composition is explained in Part 3. A description of the sound scene objects is given in Part 4, and the physical model is detailed in Part 5. A demonstration of the creative potential of the system is outlined in Part 6. Finally, in Part 7 we conclude with an evaluation of the system and a comparison between the proposed object-oriented approach and the more traditional track-oriented approach.

3 System overview

Below is given an overview of the system that decodes the XML data and renders it into a 3D sound scene.

3.1 Hardware

The reproduction system known as CHESS (Configurable Hemispheric Environment for Surround Sound) consists of sixteen identical monitoring speakers placed on a spherical mobile scaffold (Figure 1). These can be positioned anywhere around the listener. This structure allows rapid changes of speaker configuration; for instance when changing from a horizontal to a hemispherical sound field reproduction. The speakers are fed by a multi-channel sound card installed in a Macintosh G4. The studio room has carpeted walls but is not anechoic. At present, the speaker dome can be used by only one to three listeners at a time; it should be seen as a prototype of a larger dome of speakers that could accommodate a larger audience.

Figure 1: The CHESS 16-speaker dome

3.2 Software

An overview of the software architecture is depicted in figure 2. The XML encoded 3D sound scenes are parsed by Java using the Document Object Model (DOM) (W3C consortium website). The Java program performs timing and update of the scene and sends real-time commands to MAX/MSP (Cycling74 Website).

While Java controls the scene, the DSP layer created in MAX/MSP, implements a physical model that is used to calculate and simulate reflections, reverberation, delays, attenuations and finally spatialisation.

Java and MAX communicate through the Open Source Control protocol (OSC) (CNMAT website) over the User Datagram Protocol (UDP) network protocol (Ross and Kurose).

A Java 3D (Sun website) user interface is currently being implemented so that a graphical representation of the scene can be given (figure 3); this visual representation is vital when composing the scenes for getting a global overview of the scene structure and for tracking trajectories of objects etc.. In a later stage, the Java3D graphical user interface will also include authoring tools by which virtual sound scenes can be created intuitively.

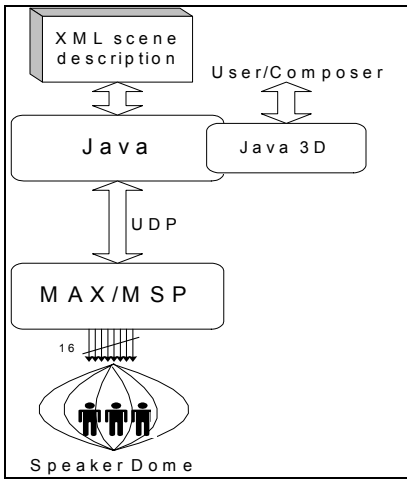

Figure 2: Overview of the proposed framework

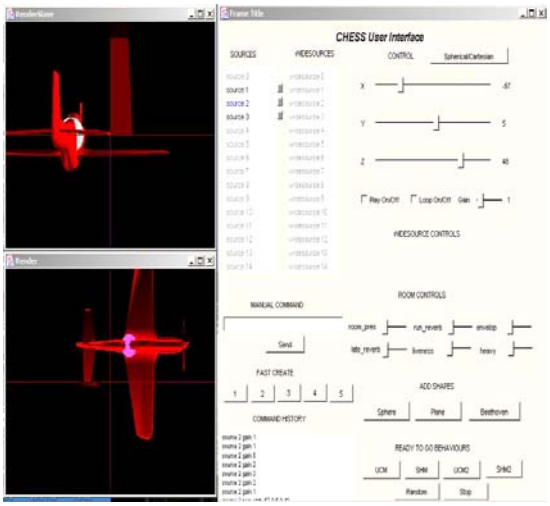

Figure 3: The Java3D user interface

4 XML scene description

For the purpose of encoding 3D sound scenes, several schemes were candidates: Virtual reality languages such as VRML and X3D (Web3D consortium) were rejected as they have only basic sound description capabilities; for instance, sound sources can only have ellipsoidal directivity patterns.

On the other hand, MPEG-4 (Peirera, Ehbrahimi 2002) has advanced 3D sound capabilities thanks to the AudioBIFS scene description language (ISO/IEC). However, implementing a fully compliant MPEG-4 decoder is a daunting task due to the high complexity of the standard.

For our application, which is describing virtual sound scenes, MPEG-4 was judged being too heavy and no scene authoring tools are available to us at present. The scene score in AudioBIFS is also not centralised; the playing time of sounds is implemented in the fields of the sound nodes, which for a scene composer, makes it difficult to have a global overview of the scene.

For theses reasons, we preferred to develop our own sound scene description scheme (Potard, Burnett 2002) that could address all our needs. The scheme had to be able to describe time varying virtual sound scenes thoroughly using a centralised scene score containing temporal information such as object trajectories and timing of events. Using a centralised scene score simplifies scene animation and re-use of the scene content. Our scene description is in some way similar to the CSound orchestra and score format. The objects in the scene form the orchestra that are in turn controlled by the score.

The 3D sound scene description scheme was implemented in the eXtended Markup Language (XML) format (Hunter 2001). XML is a mark-up language that is used in a large number of applications ranging from e-commerce databases to music notation (MusiXML). XML was selected as it goes well with the object oriented and hierarchical way of describing 3D sound scenes. Also XML is text thus human readable and many tools for parsing XML in Java are available (JDom , Xerces).

The developed 3D sound scene description scheme implements the current state of the art description features found in AudioBIFS as well as other features that are not yet available in AudioBIFS such as sound source wideness and shape description (section 5.1).

The virtual 3D sound scene scheme is implemented as an XML schema which acts as a template for the subsequent XML encoded sound scenes. The different elements of the XML scheme are now detailed.

4.1 Composition of a virtual sound scene

To construct a virtual sound scene, an orchestra and a score need to be defined. The orchestra comprises the set of sound objects of the scene. As a minimum, a scene must contain one sound source, a medium object, a listener object (the reference point) and a score. The semantics of the different objects and of the score are now given.

4.2 Sound sources

Sound sources are the sound inputs into the virtual scene. Normally sound sources emit signals generated from a monaural signal source coming from a sound buffer, a soundcard input channel or a network audio stream. The developed virtual scene description scheme also allow importing 5.1 channel and B-format recordings (Malham 1995), (Gerzon 1985) which can be regarded as complete sound scenes in

themselves. This feature is useful when constructing hybrid scenes, that is, scenes that mix recorded 3D sound environments using for instance a Soundfield microphone (Soundfield website) and additional spatialised monaural sound sources. In this case, the orientation fields of the sound source object can be used to apply rotations to the B-format recorded scenes.

Sound sources have the following parameters.

URL: Address of a sound buffer, URL of a network stream or a soundcard input channel number.

Position: Cartesian or polar coordinates of the position of the sound source in the virtual sound source. The units are meters and degrees.

Orientation: Pointing direction of the sound source expressed by a 3D vector or azimuth and elevation angles

Shape: Spatial wideness of the sound source. Using a technique inspired from a previous study (Kendal 1995), it is possible to form 'sound source shapes' made of several point sources emitting uncorrelated signals. A psychoacoustic study of this effect carried out by the author can be found in (Potard and Spille). To assign a shape to a sound source a list of points describing the shape must be entered. The shape feature can also be used more simply to create apparently wide sound sources, for instance a beach front or waterfall.

Directivity: Describe the source directivity pattern for different frequency bands. By properly setting the directivity pattern of the virtual sound source, realistic effects can be achieved. For instance a violin tends to emit more high frequency content at the front than at the back. The directivity properties are set by specifying gains at several angles. Interpolation is used so it is not necessary to set a gain per each degree. If Directivity is not specified, sound sources are omnidirectional by default.

4.3 Buffers

Buffers hold sound samples in memory. To be heard in the scene, they must be attached to a sound source object. One buffer can be shared simultaneously by several sound sources.

4.4 Surfaces

Virtual surfaces are incorporated in the scene to produce obstructions and reflections effects such as occurring with walls, furniture, persons etc..

Complex room geometries can be defined by a set of polygonal surfaces. Using this data, the early reflection pattern of the room can be estimated using an image model (Lee and Lee 1988) or ray-tracing algorithm. The surface object has the following parameters:

List of vertex points: List of points defining the corners of the virtual surface. Because of necessary complexity limitations in the early reflection calculation, only flat and polygonal surfaces can be described.

Material properties: Set of transfer functions describing the frequency and angle of incidence dependent attenuation applied on the sound ray during reflection and obstruction. The sound ray is attenuated and filtered differently whether the sound ray undergoes reflection or obstruction by the virtual surface. Different materials applied set to the surfaces help in giving a realistic representation of the scene.

4.5 Medium

The medium object defines the amount of attenuation and delay applied to the sound rays during propagation in the virtual medium. The medium is air and has the following parameters.

Speed of sound: In meters per second. This is useful to control the strength of the Doppler effect and the arrival times of direct and reflected sound rays. This can be used to exaggerate or attenuate the Doppler effect as well as making virtual rooms bigger by exaggerating delays.

Temperature, Humidity and Pressure: These parameters are fed into air attenuation equations (Harris 1966), (ISO). This is useful to re-recreate particular propagation conditions (e.g. extremely cold weather) that have quite a perceptual impact on the sound scene.

4.6 Macro-objects

Macro-objects are a grouping mechanism by which complex sound objects can be constructed. These are composed of several sound sources and/or acoustic surfaces. For example a car macro-object can be created by grouping several sound sources having specific spatial position for the tires, engine, horn and several acoustic surfaces describing the metal body of the car. This way the car-macro object will be both a sound emitting and sound field obstructing and reflecting object.

An advantage of macro-objects is that if the scene composer wants to set a trajectory to the car macro-object cited above, he/she will have to do it only for the parent macro-object and not for each individual child object present in the macro-object.

Macro-objects can be saved separately in XML format so that they can be re-imported in different scenes. We are aiming at constructing a library of macro-objects that can be readily used by scene composers.

4.7 The listener

This object describes the position and orientation of the virtual user in the sound scene. It sets the point of origin of the scene which the processing layer is based on for calculating distances, relative position of sources and reflections, etc.

This object can be set a trajectory so that the user wanders around in the virtual scene.

4.8 Scene score

The score is a separate section in the XML scene description that contains the temporal and scheduling information of the objects constituting the scene. The score has two parts: an initialisation score and a performance score.

The *initialisation score* is used to set the initial state of objects such as position of objects before the scene is rendered. The scene score format shown in figure 4 permits the utilisation of opcodes that can be used to algorithmically construct a scene. For example, a swimming pool atmosphere can be quickly constructed by using a 'splash' sound and a special opcode that distributes multiple copies of this sound source and sets random playing times to the sound samples.

The Initialisation score is composed of lines of score as depicted in figure 4. A line of initialisation score has a *Command* field containing the opcode, one or several *Object fields,* containing the references of the objects that are to be affected by the opcode (i.e. the operands), and one or several *Parameter fields* that hold parameters for the command.

The *performance score* on the other hand, describes the parameter changes of the objects over time as well as the playing times of sound sources. A line of *performance score* has a *Start time* field and a *Duration time* field, an opcode field, one or several operand fields and one or several parameter fields. The *Start time* defines when the particular action should be taken and the *Duration* field specifies the duration of the action.
Below is an example of a line of Performance Score:

5	5	MOVE	object1	3	5	6

This particular line of score results in the displacement of the object called "object1" from its current position to the position with coordinates (3,5,6) during five seconds, starting at five seconds from the start of the scene. Complex trajectories can be described by the TRAJ opcode and a long list of coordinates.

This score format is similar to the CSound score format.

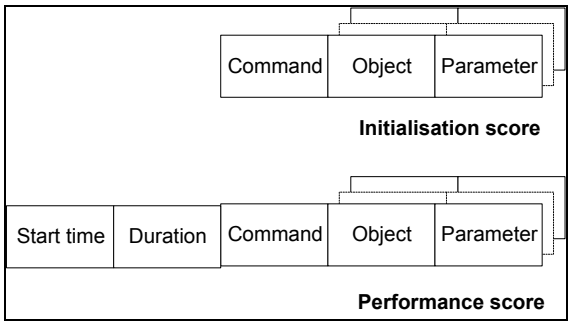

Figure 4: Format of the scene score

5 Signal Processing layer

To render the virtual sound scenes and generate the proper perceptual cues from the XML scene description, we implemented a model that uses both physical and perceptual concepts. The physical model is used to calculate signal attenuation and filtering, delays and the early reflections of the sound field. The perceptual model is used to generate the diffuse and late reverberation.

The signal processing layer is implemented by MAX/MSP using several patchers and externals (i.e. libraries) programmed in C language; these offer much better efficiency compared to MAX patchers. MAX receives real-time control commands from the Java program through a UDP network port. The Java program sends scene updates at a 30Hz refresh rate which is sufficient to obtain smooth displacements of objects. The different tasks involved in processing and placing one single sound source in the virtual scene are shown in figure 5. These processes have to be repeated for each sound source present in the scene.

Firstly, the direct signal path undergoes directivity gain attenuation, distance and air attenuation, eventual filtering caused by obstructing surfaces, some delay, shaping and finally spatialisation. The spatialised sound sources are summed onto an 'Ambisonics bus' which needs to be ambisonically decoded to obtain the speaker signals. This is useful for saving the rendered scene onto disk in a format which is independent of the speaker configuration. With Ambisonics 4th order, the bus has twenty-five audio channels.

The dry signal is also fed to an image model algorithm that calculates the positions of the visible phantom sound sources at the current time. Phantom sound sources are spawned by reflections on the virtual surfaces (Figure 8). Each phantom sound source then undergoes the same process that is applied on the direct sound signal (Figure 6) except shaping for simplicity reasons.

Finally to simulate the late and diffuse reverberation, the dry signal is fed into a

reverberation module which is controlled by the subjective parameters set in the XML description. The different modules are now detailed.

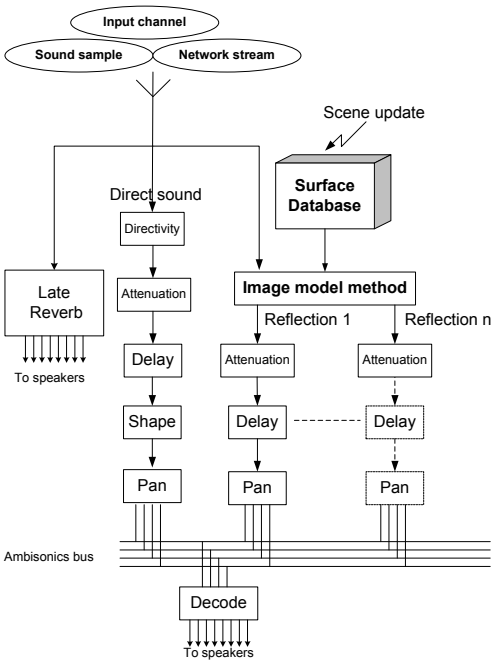

Figure 5: Overview of the processing tasks involved in the rendering of one virtual sound source

5.1 Direct sound

Simulating the direct signal emitted by a sound source requires several processing steps as shown in figure 6. First the angle between the sound source orientation and the listener position must be derived so that the proper gain is looked up in the directivity pattern description of the sound source. Then the attenuation caused by source distance and the low-pass filtering caused by air are applied. If the direct signal undergoes traveling through one or obstacles, the signal need to be filtered by the transmission characteristic of the obstructing surfaces.

Follows the shaping of the sound source; this feature is explained in figure 7, it is similar to applying a certain tonal volume to the sound source, this effect is useful for creating diffuse sound sources such as a beachfront or wind blowing in trees from a single monaural input signal. Afterwards the sound source signal is delayed by some amount of time depending on the relative distance between source and listener and the speed of sound parameter set in the medium object. Changing delays automatically create Doppler effect which is an important cue of moving sound sources (Chowning 1971). After these modifications, the signal is encoded into Ambisonics format and added onto the 'Ambisonics bus'.

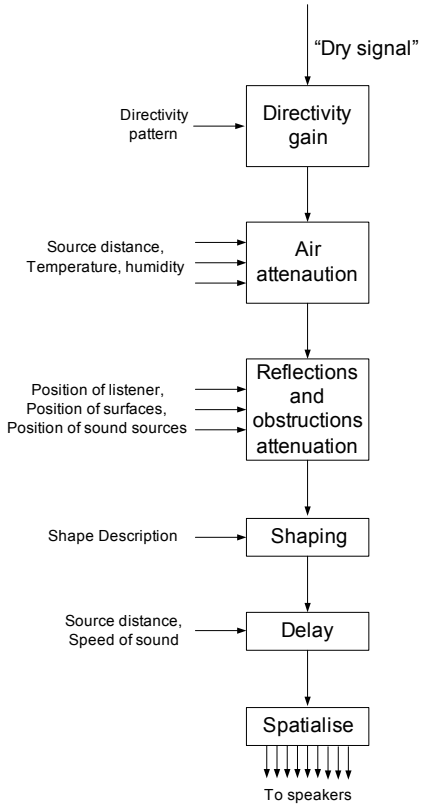

Figure 6: Signal processing tasks applied on the direct sound signal

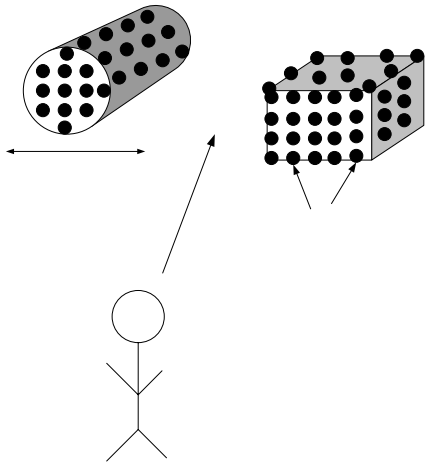

Figure 7: Illustration of sound source shaping

5.2 Early reflection calculation

From the surface database set in the XML description, an image model algorithm calculates the position of phantom sound sources (Figure 8). The number of these grows exponentially if reflections of reflections are taken into account. Therefore we have limited the early reflections calculation up the first order, which can still be computationally intensive for a large number of surfaces. The computation of

the position of the reflections/phantom sources has to be updated in case of moving surfaces and sound sources or changing listener position. The image model algorithm is implemented in a MAX/MSP C language external as it must be computationally efficient. The output of the external is a list of visible phantom sources and their position at the current time frame (1/30 second). Each phantom sound source then needs to undergo the same treatment applied on the direct sound, that is, attenuation, obstruction filtering and delay. Shaping is not applied on reflected sounds for simplification reasons.

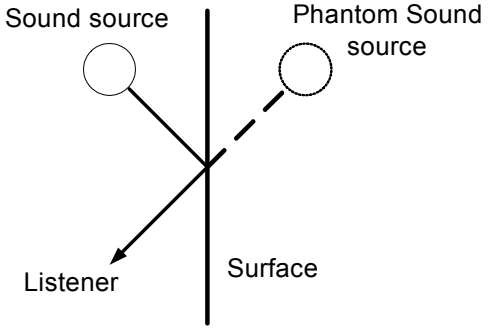

Figure 8: A sound reflection is equivalent to a phantom sound source placed behind the reflecting surface

5.3 Late reverberation simulation

It has been found that late reverberation cannot be calculated using an image method algorithm. After only few seconds the number of phantom sound sources can grow to several millions, forming the diffuse part of the reverberation. This is clearly unmanageable therefore a reverberation algorithm is used. We used a Feedback Delay Network (FDN) (Jot 1997) based reverberation that is controlled by a set of perceptual parameters set in the sound source object. This solution is used in IRCAM's Spatialisateur (Spat).

5.4 Air and surface filtering

While travelling in the medium, the different sound rays (direct and reflected) are attenuated by the medium. Air attenuation corresponds to a low pass filtering which depends on travelling distance, temperature and humidity. Air attenuation is implemented by a classic Finite Impulse Response (FIR) digital filter. Air attenuation equations (ISO) are used to calculate the frequency response of the attenuation.

Further attenuation of the sound rays when bouncing and travelling through virtual surfaces is performed in the same manner using the surface material property set in the description of the virtual surfaces.

5.5 Spatialisation

Spatialisation is the process by which the user or audience perceive the different sound sources with the correct azimuth and elevation. Distance cues however are created by the signal distance attenuation, delay, filtering and the ratio of direct sound to reverberation (Chowning 1971).

To perform spatialisation in this system we use fourth order Ambisonics (Daniel 2000), (Nicol and Emerit 1998) as it offer stable spatialisation (during head movement) and on a reasonable "sweet spot" area compared to first order Ambisonics (i.e. B-format). From Ambisonics theory (Daniel 2000), the maximum usable order is set by the number of available speakers. With 16 speakers, 4 is the maximum available order.

Ambisonics also offers great scalability as the encoded signals are independent of the speaker array configuration. Therefore in our case, the 'Ambisonics bus' can be saved onto disk and be decoded later for a different loudspeaker configuration.

Other spatialisation techniques were candidates for our system: Wave Field Synthesis (WFS) (Boone et al 1994), (Daniel 2003) is a powerful spatialisation technique but requires a very large number of speakers, especially if elevation positioning of sounds is required. Vector Based Amplitude Panning (VBAP) (Pulkki 1997) is a simple amplitude panning technique between triplets of speakers. However with VBAP we experienced low stability of the sound sources and diffuse spatialisation. Perhaps more speakers would solve these problems.

6 Creative use

The creative potential of the system described here is clearly immense. A user-interface suitable to the specific needs of musicians and composers will need to be developed if the system is to be accepted widely by the creative arts community, but even in its present form the possibilities for the construction and testing of highly complex layered sound structures are endless. It is also relatively simple to program and conceptually transparent.

Of particular interest to creative users is the potential to exploit multiple simultaneous sonic environments, highly detailed spatial trajectories, and in the creation of immersive soundscapes suited to installations in galleries and museums as well as in the concert hall. The short example that accompanies this paper sets out to demonstrate these features.

7 Evaluation

At present, composers can build virtual sound scenes by creating objects and a scene score by writing the appropriate XML description. The Java3D

interface performs decoding, visualisation and control of the scene in real time. However, for composition, authors still need to write the scene in XML format which can be a barrier to some composers. Therefore we are developing authoring tools that will permit to be completely oblivious to the XML syntax and concentrate on the scene content and its structure alone. The design of these authoring tools is decided in discussion with composers so that these answer their needs first. In this case, XML will only be used as a saving and exchange format, which can be ignored by the scene composers.

The proposed system can spatialise 24 sound sources in real-time on a Macintosh G4 800 MHz. If reflections and reverberation are added into the scene, this number drops rapidly. Latency of the whole system has not been measured but is not perceivable by the user.

The presented speaker array can only accommodate one to three simultaneous users at the moment. However, the speaker dome can be built on a bigger scale (Vennonen 1996) that can accommodate a larger audience.

7.1 Track oriented VS object oriented

The approach presented here for creating 3D sound scenes is clearly object-oriented. To compare this approach with a traditional track-oriented approach such as used in audio sequencers, we have translated some of the DSP algorithms (spatialisation, delay, filtering, reverb) into a suite of plugins using the Pluggo architecture (Pluggo, Cycling74 Website). Once created, these plugins can be inserted in a track and azimuth and elevation of the sound source associated to that track can be automated (figure 9). We have not yet implemented reflective surfaces in plugins but this appears to be problematic as surfaces would need to have their own tracks too.

While this approach is relatively straight forward and may be usable for simple scenes, many advantages of the object oriented approach are lost; for instance algorithmic scene compositions, macro-objects, and a global visual overview of the 3D scene cannot be implemented in a traditional sequencer environment.

For these reasons, we believe that an object-oriented approach is more intuitive and appropriate for the purpose of composing complex virtual sound environments since everything in the real world is made of objects.

Due to the sequencer overhead, only 3-4 sound sources can be spatialised at one time while 24 can be spatialised simultaneously in MAX/MSP on a Macintosh G4 800 MHz.

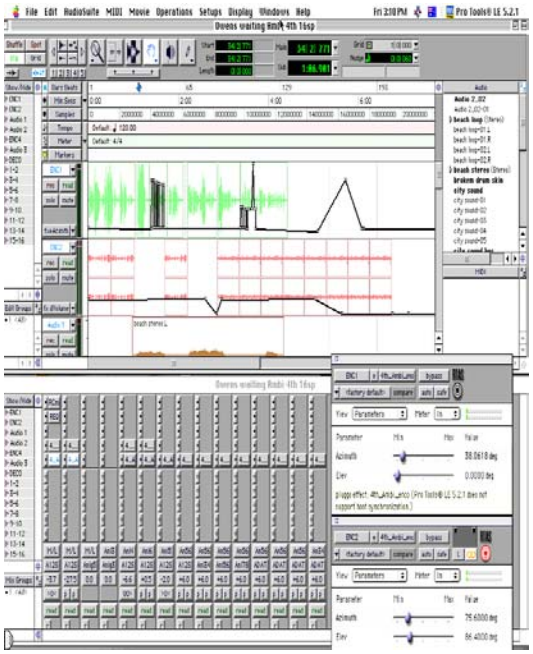

Figure 9: Composing scenes using 4th Order Ambisonics spatialisation plugins in a sequencer program

Acknowledgement

We would like to thank Didier Balez who constructed the ingenious metalwork of the scaffold, Mark O'Dwyer for his work on the Java3D interface and the Digital Media Centre of the University of Wollongong, which permitted the funding of the project.

Conclusion

A novel architecture for composing virtual sound scenes and 3D music was presented. To do so, object-oriented virtual sound scenes are encoded in XML and are rendered in real-time by a signal processing layer implemented in MAX/MSP. A Java3D user interface currently under development displays a 3D graphical representation of the scene and allows real-time interaction with the objects present in the scene. However composition of the scene is still performed in XML which can be a barrier to scene composers. Our aim is to develop comprehensive authoring tool so that composers can be oblivious to the XML syntax.

The proposed system can also be used in a live performance context as inputs from live instruments can be spatialised into the virtual scene. We are also doing early experiments with a virtual glove and 3D glasses connected to the Java3D program to provide a more efficient interface to the user than a mouse.

We believe that composition of 3D sound scenes is not properly feasible in traditional track based sequencers. We think that new tools that are object oriented based are more suitable to build virtual 3D sound scenes as advanced features such as: interaction between objects (e.g. reflections or collision), parent/child relationships, algorithmic and complex object behaviours, scene content re-use, object instancing etc. can be employed.

Ultimately, however, the real test of the proposed system and tools will be the extent to which composers and sound designers are willing to engage with both the theoretical concepts and the physical implementation outlines in this paper, and, indeed, the quality of the works produced with it.

References

Arons, B. 1992. http://citeseer.nj.nec.com/10582.html

Boone M.M. et al, "Spatial Sound-field Reproduction by Wave-Field Synthesis", Journal of the Audio Engineering Society, Vol 42(12), December 1994, pp 1003-1011

Chowning, J. "The simulation of moving sound sources". Journal of the Audio Engineering Society, 1971,19(1):2-6.

CNMAT Website, www.cnmat.cnmat.berkeley.edu/OSC/

Cycling74 Website, www.cycling74.com

Daniel, J., "Représentation de Champs Acoustiques à la transmission et à la Reproduction de Scène Sonores Complexes dans un contexte multimédia", PhD Thesis, Universite Paris 6, 2002

Daniel, J., "Further Investigations of High Order Ambisonics and Wavefield Synthesis for Holophonic Sound Imaging", Proceedings of the 114th Audio Engineering Society Convention , Amsterdam, March 2003, Preprint 5788

Essential Reality: The P5 virtual Glove http://www.essentialreality.com/

Gerzon, M.A, "Ambisonics in Multichannel Broadcasting and Video", Journal of the Audio Engineering Society, Vol. 33(11), November 1985, pp 859-871.

Harris C.M., "Absorption of Sound in Air versus Humidity and Temperature", The Journal of the Acoustical Society of America, Vol. 40(1), 1966, pp 148-159

Hunter, David, "Beginning XML", Wrox Press Inc, 2001

ISO 9613-1 Standard: 1993, Air absorption curves

ISO/IEC 14496-1 Standard, Information technology – Coding of audio-visual objects, Part 1: System

JDOM, www.jdom.org/

Jot, J.M, Warusfel O., "Technique, algorithms et modèles de représentation pour la spatialisation des sons appliqué aux services multimedia", Proceedings of CORESA97, Issy-les Moulineaux, France, March 1997.

Jot, J.M., "Efficient Models for Reverberation and Distance Rendering in Computer Music and Virtual Audio Reality", in proceedings of International Computer Music Conference (ICMC), Thessaloniki, Greece, September 1997.

Kendall, G. S., "The Decorrelation of Audio Signals and Its Impact on Spatial Imagery", Computer Music Journal, 19(4), 1995, pp 71-87,

Lee H., Lee B.H, "An efficient Algorithm for the Image Model Technique", Applied Acoustics Journal, Vol. 24 1988, pp 87-115

Malham, D. G., Anthony Myatt, "3-D Sound Spatialization using Ambisonic Techniques", Computer Music Journal, 19(4), 1995, pp 58-10

MusiXML,www.musicnotation.info/en/musixml/MusiXML.html

Nicol, R, Emerit, M., "Reproducing 3D Sound for Video Conferencing: A comparison Between Holophony and Ambisonic", in proceedings of the First COST-G6 Workshop on Digital Audio Effects (DAFX98), Barcelona, Spain, pp 17-20

Pereira, F., Ebrahimi, T., "The MPEG-4 book", Prentice Hall, 2002

Pluggo, http://www.cycling74.com/products/pluggo.html

Potard, G., Burnett I., "Using XML schemas to create and encode interactive 3-D audio scenes", Proceedings of DCW2002, Sydney Australia, April 2002, Lecture

notes in Computer science, Springer-Verlag, pp 193-202

Potard, G., Spille, J., “Study of Sound Source Shape and Wideness in Virtual and Real Auditory Dsplays”, in proceedings of the AES 114th convention, Amsterdam, March 2003, Preprint 5766

Ross, K.W. and Kurose J.F., “The TCP-Friendly Website” www-net.cs.umass.edu/kurose/transport/UDP.html

Soundfield microphone, http://www.soundfield.com/

Spat, www.ircam.fr/produits/logiciels/spat.html

Sun Website, www.sun.com

Vennonen, Kimmo, A pratical system for three-dimensional sound projection http://online.anu.edu.au/ITA/ACAT/Ambisonic/Ambisonicswork.html

Ville Pulkki, “Virtual Sound Source Positioning Using Vector Base Amplitude Panning”, Journal of the Audio Engineering Society, Vol. 45(6), June 1997, pp 456-466

W3C Consortium Website, www.w3c.org/DOM/

Web3D consortium website, www.web3d.org

Xerces, xml.apache.org/