AN AUDITORY 3D FILE MANAGER DESIGNED FROM INTERACTION PATTERNS

Christopher Frauenberger, Veronika Putz, Robert Höldrich, Tony Stockman

University of Music and Dramatic Arts Graz, Institute of Electronic Music and Acoustics
Queen Mary University of London, Department of Computer Science
Graz University of Technology, Signal Processing and Speech Communication Laboratory

[frauenberger, hoeldrich]@iem.at, v.putz@gmx.at, tonys@cs.qmul.ac.uk

ABSTRACT
This paper shows the design, implementation and evaluation of an auditory user interface for a file-manager application. The intention for building this prototype was to prove concepts developed to support user interface designers with design patterns in order to create robust and efficient auditory displays. The paper describes the motivation for introducing a mode-independent meta domain in which the design patterns were defined to overcome the problem of translating mainly visual concepts to the auditory domain. The prototype was implemented using the IEM Ambisonics libraries for Pure Data to produce high quality binaural audio rendering and used headtracking and a joystick as the main interaction devices.

1. INTRODUCTION
Audio in user interface technology has long been limited to simple beeps and alarm sounds. Despite the cases in which it was inevitable to switch from visual cues to another interaction modality, the visual screen remains to be the interface designers first choice to gap the bridge between machine and human beings. With graphical user interface design having a long and lively history resulting in vast knowledge about how to build efficient and usable interfaces. However, for a number of reasons this is about to change. The complexity of human computer interaction has reached a level at which visual-only displays sometimes reach their limits. Applications become increasingly complex offering new features to the user, but without being communicated to the user they remain unused or cannot provide their full power. Furthermore, the demands in human computer interaction changed significantly with the increasing mobility of users. Modern information technology gets integrated into our everyday life seamlessly and requires small, flexible, but powerful interface technologies. To bear with those challenges, the consideration of other interaction modalities is essential and the auditory domain is a good candidate.

Currently, audio is not an equal partner for vision in human computer interaction. Too little development has been seen in the field of auditory displays to claim that audio can equally replace visual interaction. This scientific field is mainly covered by the International Community of Auditory Displays\(^1\) (ICAD) founded 1994, a time at which visual interface techniques already were highly sophisticated. Nevertheless, the community is catching up, making auditory displays increasingly efficient and building up knowledge about how auditory communication is working. Many prototypes have proven that audio is capable of playing a major role in human computer interaction and that using it, be it in conjunction with the visual mode or as sole alternative to visual screens, is a promising approach to address the challenges in modern information technology described above.

Another aspect of using audio as interaction modality is the accessibility of computers for people with visual disabilities. As a matter of fact, computing highly dependent on graphical output is of little use for the visually impaired and blind. Relying on auditory and tactile interaction the people affected have to deal with information translated from the visual domain which basically is limited to the text available and much information presented through graphical means is lost. Screen reading software also represents the state of the art of auditory displays for computers on the consumer market. This means sequential, text orientated audio output for all what comes with a modern computer.

Given the major improvements in audio processing and all the possibilities provided by powerful sound hardware as well as software, this paper argues for using all those audio technologies in conjunction with HCI methodologies to improve the quality of auditory user interfaces. We have built a prototype of a file managing application using a spatial auditory rendering system to set up a virtual 3D environment as a user workspace.

Graphical design has produced a number of reliable concepts for robust and usable user interfaces and we propose to learn from these methodologies for the auditory domain while keeping in mind that there are significant differences in the way either domains communicate. This was the reason for adapting the well established pattern design method for our prototype by introducing a mode independent meta domain. With this concept user interfaces can be designed without determining their means of realisation and specialised transformations can be developed for each interaction domain (e.g. auditory or visual). With that, the strengths of every domain can be preserved and while using a method well known from the graphical design, we do not apply graphical concepts on auditory interaction.

The following section describes the current state of the art in audio rendering and auditory display development. Subsequently, section 3 shows the design method used and introduces the design process with mode independent interaction patterns. Section 4 describes in detail the prototype developed, its design, its implementation and the evaluation test conducted. Finally, section 5 concludes the paper and provides an outlook on future work.

2. STATE OF THE ART

The increasing computational power available for digital signal processing has made increasingly complex simulations of acoustical environments possible. The simulation of acoustical scenes

\(^{1}\)http://www.icad.org

DAFX-1
with sound reproduction techniques create natural environments which are customisable and controllable in real time, very much as visual virtual environments were developed [1]. Acoustical rendering of objects (sound sources), the environment and the listener can be realised using a number of different approaches like Ambisonics [2], Wave Field Synthesis [3] or Vector Based Amplitude Panning [4]. All of those techniques are capable creating the sensation of various output formats ranging from large scale loudspeaker arrays to binaural rendering for headphones. This technology was just recently introduced to create auditory displays as a major step towards natural interfaces [5, 6].

A variety of auditory displays were developed for specific problem domains (e.g.: [7, 8]) and some efforts were taken towards a structured approach for more generic solutions. Early proposals include the Mercator project, the first framework targeting customary Unix desktops [9]. Another proposal was Y-Windows also following the idea of building alternative, audio rendering engines (servers) for existing clients requesting their user interface representation [10]. However, both approaches implied that graphical concepts were translated into the auditory domain and therefore had their limitations. A first attempt to depart from this approach and introduce a mode independent meta domain was made in [11] and subsequently led to the concept proposed in [6, 12] and in this paper.

Attempts to employ usability engineering methodologies in the design of auditory displays include the investigation of audio metaphors [13] and other structural approaches to include sound into human-computer interaction (Earcons [14]). Recently, the proposal of using patterns in sonification highlights the advantages of such methods in re-usable designs [15, 16].

However, much more may be exploited from the discipline of usability engineering. Auditory representations of user interfaces are in need of profound heuristics to assess user satisfaction similar to those in the graphical domain [17]. Examples of where work is needed to identify heuristics to guide the process of auditory display design include minimising the problems incurred due to the transient nature of sound, quantifying how the effectiveness of interactions can be improved through learning and providing guidelines for how sound can best be integrated with other media [18]. Also the design pattern method is a promising approach and other design principles may also give more control over the efficiency of auditory displays.

3. PATTERN DESIGN

Interaction pattern design has been successfully used to create robust user interfaces in the visual domain. The main advantages of this method can be summarised by

- re-usable solutions to common HCI tasks
- consistency of solutions in different interfaces
- easy and quick way to build interfaces from scratch.

There are various sets of patterns available with different objectives and at different levels of abstraction. To achieve the goal of finding a common base for HCI in different modalities we evaluated sets of patterns and found Welie and Trefteberg’s patterns to be the most suitable for our purposes [19].

We used a reformulated set of those patterns based on the presentation of a user interface. These objects may result from one or more interaction patterns transformed into the representation medium. Despite these small changes to the terminology, the existing interaction patterns were easy to re-formulate so that they would not prejudice the process of realisation.

While developing the patterns, we recognised that certain tasks or parts of patterns recur in other patterns too. This led to the concept of atoms and contextual attributes. Similar to a vocabulary for instantiating designs, a set of atoms were developed from which patterns may draw when addressing a particular set of user requirements. This also implies consistent representation of similar elementary units throughout the whole interface although atoms are not sufficient to solve any interaction problem. In order Not to end with a totally unrelated patchwork of small pieces of a user interface, each atom provides contextual attributes. These attributes need to be set by the parent pattern in order to indicate their context. In the graphical domain this would, for example, mean that certain elements like buttons or text fields are in the same window sharing the same frame and background colour. The following contextual attributes were identified for our set of atoms:

Similarity: Atoms in the same pattern share properties like timbre, rhythm or type of voice in their acoustical representation.

Proximity: Atoms in the same pattern are grouped based on the available dimensions of the representation area (space or pitch ranges).
Homogeneity: The same types of atoms should be placed adjacently in a pattern on the basis of the available dimensions of the representation area (space or pitch ranges).

It is important to state that not only the patterns and the atoms undergo the transformation process in order to form a real user interface, but also the contextual attributes must be mapped into the different representation media. Their realisation in the auditory domain will differ considerably from the visual domain.

The subsequent section shows the design and realisation of an auditory version of a file management application using a subset of these patterns. It would be out of scope of this paper to describe all patterns developed in great detail so here we shall only show the patterns used for the Audio Explorer. The full set of patterns and more information about them is available from [20].

4. THE AUDIO EXPLORER

To prove the concept of design patterns with auditory displays we implemented a prototype display of a real world application. Being fundamental to every operating system and fairly well known to most computer users, a file managing application was chosen for that matter. The goal was to analyse the existing graphical application, describe it using the set of mode-independent patterns and then create an auditory display with that description.

4.1. Design

Analysing the Microsoft Explorer application resulted in a description using the following patterns (atoms):

- Container Navigation (tree structure, list)
- Command Area (triggering element, selection)
- Context Menu (triggering element, selection)
- Message (triggering element, raw information)

Although not the whole functionality of the MS Explorer was considered in this analysis, the prototype was still covering the most important functions for a file manager.

The Container Navigation pattern was used to describe the two main frames of the Explorer. For the folder tree in the left frame the tree structure atom was used and the list atom described the right content frame. The Command Area pattern described the menu structure and the tool bar area. Finally, the Contextual Menu pattern solved the availability of the context menu and the Message pattern was used for all pop-up windows at their occurrence. Figure 2 shows the basic layout of the virtual audio environment into which the patterns were transformed. The container navigation pattern was realised as two different areas in the virtual environment, the walls to the front and to the right. The listing was put on the front wall using speech for the name of the items with different voices to indicate the type (folder or file). The tree structure was laid out on a grid on the wall to the right with the left-bottom corner being the root and the right-top corner the last hierarchical level. Unfolding and folding a node in the tree was also implemented. In both areas the user was able to select items and get a context menu. The contextual pattern was realised as sticky objects following the user wherever she moves. The content of the contextual menu pattern was again solved by triggering elements. The same concept was used to realise pop-up windows - sticky objects remaining to the front of the user, but with different background sound. The menu was created on the left wall of the room using speech sources lined up along the wall. When clicked they unfolded their content towards the ceiling and a series of raising tones indicated the number of menu items. The user could then virtually move vertically to select the item desired.

All sound used incorporated speech and non-speech sound and was audible within a certain range. This means that there was silence in the starting position, but moving towards a wall meant that one could hear the 5 menu items at once at different levels depending on the distance. Interaction with the prototype was done by joystick and keyboard. To avoid confusion while navigating through the virtual environment no relative movements are supported. Bringing the joystick to the starting position means moving to the centre of the room facing the front wall. Moving up, along the z-axis, in the environment was implemented using the throttle handle of the joystick. The localisation of different sound sources was improved by using a headtracker.

4.2. Implementation

The virtual environment was implemented in Pd\(^2\) using binaural rendered Ambisonics. The Ambisonics libraries developed by the Institute of Electronics Music and Acoustics Graz as an extension to Pure Data were used to simulate the environment in a efficient and flexible way [21, 22].

With this system, Ambisonics is used as an intermediate stage towards a final binaural mixing. First, all sound sources are encoded in Ambisonics resulting in a number of Ambisonics channels. In this domain the encoded sources can easily be mixed and rotated before Ambisonics decoding calculates a set of corresponding loudspeaker signals. For binaural rendering these signals are then convoluted with an appropriate set of head related impulse responses (HRIRs). In order to improve the external localisation of sound sources and the naturalness of the virtual environment, there was also a room model implemented considering early reflections and late reverberation. The described Ambisonics rendering system has a number of advantages that makes it especially suitable for the task given:

- Very efficient for a large number of sound sources. This is particularly important as all mirror sources resulting from the room model are treated the same way as original sound sources, increasing the amount of sources significantly.

\(^2\)Pure Data by Miller Puckette
• Efficient support for rotation in the Ambisonics domain with rotation matrices. Using headtrackers this is an important requirement.
• Easy adaptation of the output format. This rendering system can be used with either binaural output or multi-channel output for an array of loudspeakers.
• Efficient for binaural output because the number of HRIRs needed is independent from the amount of the sound sources and no interpolation between HRIRs needs to be calculated.
• Incorporated room model with early reflections and late reverberation supporting the externalisation of sound sources.

The used library consists of six modules which contain objects and abstractions to be used with Pure Data. iem_matrix provides efficient matrix operations in the signal domain and is basically a collection of helper functions. iem_ambi contains all objects for encoding, decoding and rotation in the message domain meaning it is providing all necessary parameters to the objects that actually handle the signals. iem_bin_ambi is the module responsible for decoding Ambisonics signals to a virtual loudspeaker setup and applying HRIRs to the decoded signals. It currently uses non-personalised HRIRs from KEMAR as well as CIPICS databases, but may well be used with any others too. Because this module requires the most computational effort in the whole rendering process, a number of optimisations were developed to make this module highly efficient. These optimisations include:

• Reduced HRIR set: due to the symmetry in the virtual loudspeaker setup the number of HRIRs could be reduced to its half.
• Frequency domain filtering improves computational efficiency compared to time domain implementations.
• Reduced IFFT: By combining the decoder and HRIR filters and transforming them into the frequency domain reduces the number of required IFFTs to two, one for each ear-signal.

iem_roomsim provides the room simulation with calculation of early reflections of first and second order. iem_reverberation implements a computational efficient calculation of a reverberation algorithm based on former work of J.-M. Jot and A. Chainge and abstractions to be used with Pure Data. iem_ambi

iem_ambi

iem_bin_ambi

iem_binary

iem_matrix

iem_roomsim

iem_reverberation

iem "
of an auditory file-manager application using the IEM Ambisonics library and described the evaluation process of the resulting prototype. The quality of the auditory realisation clearly met the requirements while flaws were detected in the realisation of certain design patterns. However, the concept made it possible to isolate the usability problems and assign them to certain patterns or atoms in design. This makes it possible to improve smaller pieces of an interface instead of dealing with the whole.

Future work will need to improve the patterns to be consistent and complimentary. This will need further investigations in psychoacoustics, acoustic communication, sound design, aesthetics and navigation and orientation in virtual audio environments. On the concept side, work needs to proceed towards formalising the patterns and the development of supporting tools in order to support user interface designers.

6. ACKNOWLEDGEMENTS

We would like to thank all who volunteered to participate in the evaluation test. Especially the people from the ISIS group (Integration, Service, Information and Schooling) in Graz / Austria for the close collaboration and for giving us an idea about the needs the visually impaired have when it comes to interaction with computers. ISIS is a project run by BFI (Berufsförderungsinstitut) Styria.

Support by the SonEnvir (http://sonenvir.at) project sponsored by the Zukunftsfonds Steiermark is gratefully acknowledged.

7. REFERENCES


