

# Layered Software Architecture for Designing Environmental Sounds in Non-Visual Interfaces

A. DARVISHI, E. MUNTEANU, V. GUGGIANA, H. SCHAUER  
Department of Computer Science (IfI), University of Zurich, Winterthurerstr. 190, CH-8057 Zurich, Switzerland,  
tel. +41-(1)-632 7098, e-mail: darvishi@ifi.unizh.ch

M. RAUTERBERG  
Usability Laboratory, Work and Organizational Psychology Unit  
Swiss Federal Institute of Technology (ETH), Nelkenstr. 11, CH-8092 Zurich, Switzerland

M. MOTAVALLI  
Swiss Federal Labs for Material Testing and Research (EMPA), Uberlandstr. 129, CH-8600 Dubendorf, Switzerland

**Abstract.** This paper presents a partially implemented layered software architecture for describing and designing environmental sounds (everyday sounds) in non visual interfaces based on a new sound model (audio framework). This architecture facilitates non-speech audio presentation of objects and interaction mechanisms to non visual interfaces. Physical layer, system sound software layer, sound analyser/synthesiser layer and interface layer define the different layers of this new architecture. The suggested sound model (audio framework) can be used as a basis for the description and design of environmental sounds in non visual interfaces. This paper will describe its different components: physical modelling, interaction, context sensitivity, and metaphorical description. The term *audio framework* will be used ultimately for the presented sound model. This paper will first give an introduction to the background on the use of computers by blind operators and the development of graphical user interfaces and their impact on this group of computer users, the use of sounds in CAL applications for visually impaired users and two examples for adaptation of GUIs for blind computer users, existing approaches for modelling environmental sounds, and then presents the new audio framework and its implementation in a layered software architecture. Finally some implemented sounds are discussed.

**Key words:** visual impairment, non visual interfaces, auditory interfaces, sound design, sound synthesis, sound models, human computer interaction, auditive feedback, usability engineering

## 1. Introduction

For much of their history, computer displays have presented only textual and numeric information to their users. One benefit of this character-based interface was that users who were blind could have fairly easy access to such systems. Users with visual disabilities could use computers with character-based interfaces by using devices and software that translated the characters on the screen to auditory information (usually a synthesised human voice) or/and tactile terminals and printers. Since the mid 80's, the computer industry has seen a remarkable increase in the use of GUIs, as a means to improve the bandwidth of communication between sighted users and computers. Unfortunately, these GUIs have left a part of the computing population behind. Presently GUIs are all but completely inaccessible for computer users who are blind or severely visually-disabled.

### 1.1. Use Of Sounds in CAL (Computer Aided Learning) Applications for Visually Impaired Users

Sounds such as the following are being used in different CAL applications for visually impaired users: reading technical documents and Algebra [1]; teaching blind students at schools [2]; providing access to chemistry and other laboratories [3]; revealing document structures [4]. In all of these applications the sound parameters such pitch, duration etc. are mapped to audible information which are not easily available for these group of users.

### 1.2. Using Sounds In GUIS Adaptation For Blind Computer Users

The following two projects were recently completed and describe two major efforts to adapt GUIs for blind computer users and are going to be commercialised:

- As a first example the Mercator project [5] aims to provide access to X-Windows and UNIX workstations for computer users who are blind or severely visually impaired. The interface objects (such as icons, windows etc.) are organised in a hierarchical tree structure which can be traversed using a numerical key path. The primary output modality is audio (synthetic speech and non speech audio) and, recently, Braille output. The interface objects in the Mercator environment are called AICs (Auditory Interface Component). The type and attributes of AIC-objects are conveyed through auditory icons and so called "Filtars". Auditory icons are sounds which are designed to trigger associations with everyday objects, just as graphical icons resemble everyday objects. An example of some auditory icons are: touching a window sounds like tapping on a glass pane, searching through a menu creates a series of shutter sounds, a variety of push button sounds are used for radio buttons, toggle buttons, and generic push button AICs, and touching a text field sounds like an old fashioned typewriter.

- The GUIB (Textual and Graphical User Interfaces For Blind People) is a second example [5]. GUIB uses concepts and methodologies of visual interfaces and translates them into other modal sensors primarily Braille and also audio. Some special devices are built for input and output. Spatial 3-dimensional environments and auditory icons are also integrated. The results of this project is partially commercialised in a product called "Windots".

### 1.3. Current Approaches For Synthesis Environmental Sounds

The first approach (event oriented approach) is used in the work of William Gaver [6]. It is based on the perception of events in the real world and uses the results of protocol studies and semi-physical considerations of objects (wood, metal etc.) to derive parameters for sound synthesis.

Our approach [7] (interaction oriented approach) is based on the perception of the interaction of objects in the real world and analyses the real impact sounds and concurrently encounters physical modelling of interacting objects in order to derive appropriate parameters for sound synthesis. The recorded impact sounds were analysed via spectral analysis.

## 3. Audio Framework: a Model for Describing Environmental Sounds

In this section and the following subsections different components of the new model are introduced. We have focused first on the sound generation of *impact sounds*, in particular the interaction of different spheres and beams. The interaction of these objects are analysed (sound analysis) and implemented (sound synthesis) on a SIG-Indigo workstation in the programming language OBERON (object oriented programming language).

### 3.1 Physical Modelling

We started with describing the physical models for simple interactions, e.g., the collision between a homogeneous and isotropic sphere and a homogeneous and isotropic plate/beam. Through physical modelling we are able to calculate the natural frequencies of the interacting objects and the shapes and initial amplitudes of the natural frequencies of the hit objects.

### 3.2. Interaction

Every sound can be described as the result of one or several interactions between one or several objects at a specific place and in a specific environment. Each interaction has attributes, that influence the generated sound. Also, the participating objects which take part in the sound generating process can consist of different physical conditions (states of aggregation). Various materials also have different configurations. The materials themselves also have attributes, influencing the generated sound.

An example for an interaction specific parameter is the height from which the sphere falls. Another example is the radius of the sphere which hits the beam. The bigger the sphere which hits the surface the louder the perceived sound. These two examples emphasise the importance of interaction parameters in sound producing events. Both interaction parameters are implemented in our system.

### 3.3. Context Sensitivity

Sounds are context sensitive, i.e., the generated sounds differ depending on the environment where the interaction of objects takes place and the combination of interacting objects (the same sphere hitting a wood beam sounds different from hitting a steel beam). Other examples for context sensitivity parameter are the spatial co-ordinates of impact place. If the plate/beam is hit in the middle it sounds different from when it is hit on the edge.

### 3.4. Sound Metaphors

Basically, there are three ways of describing everyday sounds on a metaphorical level; (1) linguistic description of everyday sounds which often is ambiguous; (2) Technical description of environmental sounds in terms of frequency, duration, timber etc.; (3) Semantic description of environmental sounds in terms of interacting objects, interaction and environment. Sound generated by our model fall into this last category. Further user tests are necessary to insure the adequate identification of the sounds.

## 4. Layered Software Architecture for Designing Environmental Sounds

This layered software architecture is partially implemented and supplies tools for automatic analysis and synthesis of environmental sounds.

### 4.1. Physical Layer

This layer builds the required hardware for sound recording (Microphone, DAT, AD-converter etc.), sound processing (storage, CD-driver etc.) and sound generating devices (DA-converter, loudspeaker etc.).

### 4.2. Sound System Software Layer

This layer builds the interface to different sound hardware and supplies basic procedures for sound processing (software library).

### 4.3. Sound Analyser / Synthesiser Layer

This layer consists of several units (software tools) which fulfil different tasks. These units (software tools) are:

- *Wave form editor*. Represents the set of samples for each channel (for stereo signals) on the screen and has all the standard functions of a standard editor, like *copy*, *paste*, *cut*, *save*, etc., as well as functions specific to our purposes (zoom, play, etc.)

- *Spectral analysis*. The software makes the Fourier transformation of the natural sounds by means of fast algorithms. One can specify the desired sound fragment to analyse in terms of the number of samples or in terms of time moments. In order to observe the spectrum's time evolution a procedure has been implemented that makes and draws a so-called spectrogram of the signal.

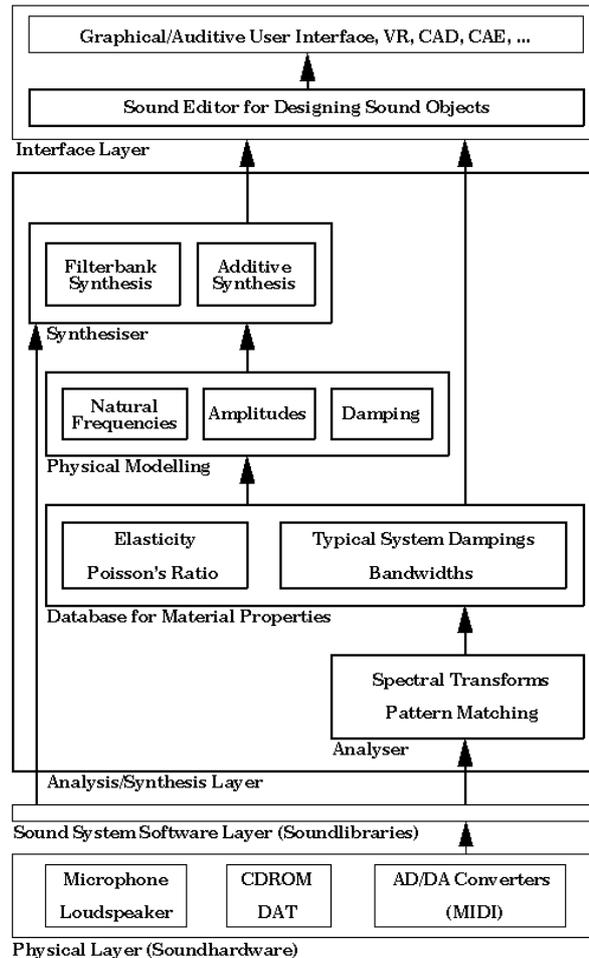


fig. 1 The architecture of a layered software system for implementing the new model

- *Parameter extraction.* There are implemented fast algorithms that sweep the time axis and/or frequency axis and find the frequencies corresponding to the natural modes of vibration, the initial amplitudes for each of this waves, and the damping coefficients that describe the spectrum's evolution in time domain.
- *Data base system.* This data base system stores two type of information; (1) different material properties (2) a formal description of system damping of different materials which are derived from analysis of the real sound of designated materials.
- *Sound synthesis objects.* Different objects are provided for sound synthesis. Currently we have implemented additive synthesis and filter bank algorithms for the synthesising of impact sounds. However, new objects (other synthesis algorithms) can be added to this layer. The employed synthesis algorithms receive natural frequencies of the vibrating objects and their initial amplitudes and the damping function for each frequency.
- *Physical modelling.* The unit for physical modelling of the interacting objects takes as input the object definition from a graphic editor (see interface layer) and generates as output different natural frequencies of the vibrating objects and their initial amplitudes.

#### 4.4. Interface Layer

This layer defines the interface to the user (software-developer) and offers an interactive editor for the definition of interacting objects, e.g., object type (wood, metal etc.), object shape (beams, plate etc.), and the environment (room etc.). The editor generates as output a meta-description of the desired sounds which can be linked to a programming environment or which can be executed (real-time sound generation). The output of this layer is used as input for the underlined layer (sound synthesiser layer).

## 5. Implemented Sounds

Different impact sounds such as bouncing metal, wood, PVC, glass and other materials have been realised and tested. One could specify these sounds in terms of their parameters (see section 4.3.). They are fully physically modelled. Other type of impact sounds are supposed to be implemented. Examples of implemented sounds can be heard on World Wide Web. (<http://www.ifi.unizh.ch/groups/se/sound/SoundProject.html>)

## 6. Conclusion

Audio Frame Work a model for the description of environmental sounds was presented. A layered software architecture based on this model is also discussed. This architecture supplies the basis for applying these sounds in GUI adaptation and potential new CAL applications. The introduced architecture allows not only tools for sound synthesis but also sound analysis. Use of environmental sounds makes non visual interfaces more intuitive for blind computer users. Further investigation is needed to find out the potential and restrictions of this approach.

## References

- [1] T.V. Raman, D. Gries, Interactive Audio Documents. In: Proceedings of ASSETS`94. ACM ISBN: 0-89791-649-2, ACM Press, Los Angeles, 1994, pp. 62-68.
- [2] D. Burger, C. Mazurier, S. Cesarano, J. Sagot, The Design of Interactive Auditory Learning Tools. In: Non-Visual Human-Computer Interactions, ISBN 2 7420 0014 3, Colloque INSERM/John Libbey Eurotext Ltd. vol. 228, Paris, 1993, pp. 97-114.
- [3] D. Lunney, R.C. Morrison, Auditory Presentation of Experimental Data. In: E. Farrell (Ed.). Extracting Meaning from Complex Data: Processing, Display, Interaction. Proceedings of the SPIE, vol. 1259, pp. 140-146.
- [4] S. Portigal, T. Carey, Auralization of Document Structure. In: Proceedings of International Conference of Auditory Display, Santa Fe, New Mexico, 1994.
- [5] W.K. Edwards, E.D. Mynatt, K. Stockton, Providing Access to Graphical Users Interfaces- Not Graphical Screens. In: ASSETS`94, Marina del Rey, CA, USA, 1994, ACM ISBN 0-89791-649-2, pp: 47-54.
- [6] M. Beaudouin-Lafon, W. Gaver, ENO: Synthesizing Structured Sound Spaces. In: Proceedings of the ACM-UIST`94, Marina del Rey, CA, USA, (1994), ISBN:0-89791-657-3, pp: 49-57.
- [7] Darvishi, A., Guggiana, V., Motavalli, M., Munteanu, E., Rauterberg, M., Schauer, H. " Designing Environmental Sounds as Result of Interaction Between Objects in the Real World" In Proceedings of INTERACT`95, Lillehammer, N, (1995, in press).
- [8] Darvishi, A., Guggiana, V., Munteanu, E. Motavalli, M., Rauterberg, M., Schauer, H. (1994) "Synthesising Non-Speech Sounds to Support Blind and Visually Impaired Computer Users" In: 4th Int. Conference on Computers for Handicapped Persons, Vienna, pp. 385-393.
- [9] Darvishi, A., Munteanu, E., Guggiana, V., Motavalli, M., Rauterberg, M., Schauer, H. (1994) "Automatic Impact Sound Generation for use in Nonvisual Interfaces" In Proceedings of 1st Annual International ACM/SIGCAPH Conference on Assistive Technologies, ACM ISBN 0-89791-649-2, pp. 100-106.