Sound design for an auditory reproduction of a graphical user interface

Institute for Perception Research
P.O. Box 513 - 5600 MB Eindhoven

15.05.1995

Rapport no. 1053

Sound design for an auditory reproduction of a Graphical User Interface

J. Fruman

Sound design for an auditory reproduction of a Graphical User Interface

J. Fruman

Abstract

Since several years, there is an increased use of graphically oriented user interfaces, in computer systems that are present in office environments. Visually impaired people can only work with these interfaces, if an intermediate system translates the visual information into a form, that these people can use. Such a system is currently being developed, as part of a project, at the Institute for Perception Research. The goal of the project is; providing visually disabled people with access to GUI's, while maintaining as much of the specific GUI aspects as possible. Providing access to a GUI is accomplished by translating the visual information into the auditory domain. For this purpose either speech, non-speech audio, or combinations of both, is used. This paper describes the development of the non-speech sounds, that will be used in the alternative GUI representation. The general aspects of the use of sound are discussed and some guidelines for the production of sounds are extracted. Successively, the sound sets that are created according to the guidelines, are listed and followed by some expectations about their use in practice. At the end of the paper, useful suggestions, for the use of sound in representing interfaces, are provided.

1 Introduction

For many years the auditory feedback the computer provided, was limited to some simple 'bleeps'. Although the visual user interface has undergone major leaps in development during the years, the auditory part stayed behind and remained underdeveloped. Today, interfaces are still mainly visually oriented, but in some fields, an auditory representation as an alternative to the visual one, receives increased attention.

Most computer interfaces used in office surrounding consist of a graphical environment that contains the interface-objects. Interface-objects can for instance be windows, push-buttons or menus. To interact with the computer, the user can manipulate these objects. A standard input-device for this purpose is the computer mouse. By moving this device, the user can move a pointer over the screen. With this pointer the objects, displayed on the screen, can be manipulated.

When sounds are used in an interface, they are often only meant to be an extension to what is already being displayed visually. However, for visually impaired people, the use of audio to represent a computer interface can be essential to be able to work with the computer at all. For blind people, the actions with and through the pointing-device are difficult to perform. They need some kind of intermediate system, that gives them access to the Graphical User Interface (GUI). Over the years, several systems with this purpose have been developed. If it is supposed that the user should process all information in the auditory domain, an interface that is entirely based on sound is necessary. Such an interface is currently in development at the Institute For Perception Research. The main goals of this project are (Poll4):

1 Providing the visual disabled people with GUI admission, so they can share the same resources with their sighted colleagues.

2 Making this sharing of resources possible, the non visual admission should be as transparent as possible. Most of the original GUI aspects should therefore be maintained, like the way the user can interact with the system. In this perspective, the mouse should keep its function as standard input device,.

3 Giving the user the impression that he or she actually works with a GUI, instead of only having admission to it.

4 Providing a spatial organisation of the non-visual objects is required, to be able to maintain the typical GUI features. An example of such features is the possibility to manipulate objects by, for instance, dragging them.

5 Presenting all the information of the interface to the user, either by speech, or non-speech sound.

The way in which the mouse as input device can be retained (point 2), is by using absolute, instead of relative positions. This is possible by using a platform, with upstanding boundaries, that represents the screen dimensions. The physical position of the mouse on this platform will correspond to the actual position of the mouse pointer on the screen. The resulting interaction device, which is a combination of this mouse-platform and the sounds provided by the system, is called the SoundTablet.

To test the practical efficiency of the basic presentation principles of such an approach, some exploratory experiments have already been conducted. The results of these tests indicated that the basic concepts of the system are a good fundament for further development. Also most test subjects indicated that they highly appreciated the approach that was chosen for this project. (Poll4)

Now steps have to be taken to come closer to a real windows operating environment. All the windows objects and events for the alternative interface should be implemented and provided with appropriate sounds. The sounds used so far, were chosen on a rather intuitive base. No special effort was made to examine more specifically what sounds can be used best for the auditory representation of the windows interface.

In this perspective, the Sonology department of the Royal Conservatory in The Hague was contacted. At the institute for sonology, subjects that have to do with electronic music and sound synthesis are educated. With the available knowledge and practical experience on sound synthesis at this institute, the collaboration should lead to the design and implementation of sounds, that can be used in the experiments with the auditory interface.

This paper is a result of this collaboration. It starts with a description of the status-quo in the field of auditory interface design (chapters 2 and 3). These two chapters are a preceding to the process of putting up guidelines (Chapters 4 to 9) for the implementation of the sounds that should be used (Chapters 10 to 15). The paper is concluded by a discussion, that includes some suggestions for the further continuation of the project.

The problem of presenting a GUI to the blind is still rather recent, because the introduction of such interfaces dates from about 1980. Much of the knowledge presented in this paper is collected from literature, that dates from about 1986 untill the end of1994. Hopefully, the experiments that will be conducted with the sound sets, that are described in this paper, will confirm the usefulness of the proposed guidelines and yield some new information that can be of use for future sound designs.

2 Using Sound in Human Computer Interfaces (HCI):

2.1 Reasons for using sound in computer applications:

There are certain fields in which the auditory representation of otherwise visually displayed items, receive increased attention. This is especially true for programs used in data presentation and analysis applications. Examples are programs for the analysis of seismographic-, stock market- , scientific- and economic data. Some underlying reasons for the increased use of audio information in these fields are:

Sound can be simultaneously processed with the visual data. In this way it can provide additional information or serve as an enhancement, to the data that is visually being presented.

Sounds can be heard anywhere. You do not explicitly have to focus attention at the sound source. This in contrast to visually displayed information. You have to watch the screen, to be able to see the visual information.

In certain cases, trends in data-bases that are difficult to be detected visually, are more easily detected audibly. For instance, a graph that represents certain data could look like a straight line. However, when the data-values are represented by the pitch of a sound, variations in pitch could indicate that the line is not as straight as it visually appears to be.

When parameters of a data-set are coupled to parameters of the sound generating system, it is possible to provide the user simultaneously with more dimensions, than the about 2.5 dimensions that can be visually displayed. For example the sound properties pitch, amplitude, rhythm and spectral contents, could each represent a certain aspect of the data set. These parameters should be attached to the data set in a sensible way. This means that the parameters have to be as independent as possible. In this example, already four dimensions of the data set are then represented.

By making use of specific techniques for combining sound-parameters, useful applications for comparing (nearly identical-) data sets are possible (Scaletti).
Nearly identical data-sets can have almost the same shape when presented visually. Small differences are then difficult to distinguish. By combining the auditory representations of these data-sets in certain ways, these difference can cause easily detectable changes in the resulting sound output. You might say that the differences are in a way, magnified in the auditory domain.

The sound output enables the user to monitor continuous processes, even processes that run in the back-ground, or that are the result of actions of other users in an multi-user environment. In this way the sound output will represent information, that is not directly visible (Gaver2), (Scaletti), (Kramer2).
If in such an environment, important trends in the processed data develop or important events occur, the user can be alerted. The user will become aware of the changes because he/ she will be able to detect or recognise a trend, or because an additional alerting sound cue is being played.

The resolution and the information processing capacity of the auditory senses are less sophisticated than that of the visual senses. Still the auditory system is the next most complex system to be considered for the processing of information. If a high density of information should be passed on to the user, making use of the auditory senses seems more appropriate, when compared to other sensory mechanisms, like touch and smell.

2.2 Terminology:

In the development of the use of audio-information in computer interfaces, several approaches can be distinguished. What is necessary, is a general classification of sounds and their aspects. Such a platform can then serve as a good fundament, where the more specific sound designs can be based on. In every different approach, such fundaments are laid down in a model.

Previous external experiments and already existing systems show that there are several models possible, for translating the visual information into sound. Differences in models, strongly depend on the kind and the intended use of the applications. For instance, if sounds are used to represent data, the priorities are totally different, when compared to using sound for representing an entire interface structure. Many of the models come with their specific terminology . All of the used terms have their own specific properties. For a better understanding, these terms will be discussed here, together with a description of their properties and common use.

Audification:

Audification is a direct translation of data series into the audible domain, to be able to monitor and comprehend them. The data itself is therefore shifted into the audible domain, by converting it to an analogue signal and amplifying it (Kramer2).
For example, a data-set of 44100 samples, could be played within one second. The result will then be a wave-form that is played back in CD-quality. The kind of sound that will be produced, depends on the properties of the data. These properties will determine if the resulting sound will be just noise or perhaps a pitched sound with a very specific timbre. If necessarily the sounds can be looped, so they can be sustained.

0th-order mapping:
Is the same kind of mapping as audification, thus the data stream itself is listened to as a stream of audio samples. Audification, or 0-th order mapping, has the fewest applications in auditory data representation. Most data sets that suit this kind of representation meet the following criteria:

They represent a single time-dependant phenomenon (or a phenomenon that can be decomposed into several one-dimensional time-dependant processes).
They are likely to be periodic or quasi-periodic.
They are relatively large sets (Scaletti).

Sonification:

Sonification is the use of data to control a sound generator for the purpose of monitoring and analysing this data. There are substantial mediating factors, as the sound generation technique needs not to have any direct relationship to the data that is provided (Kramer2).

1st-order mapping:
Is just another name for sonification, where the data stream controls parameters of a synthesis model.

2nd-order mapping:
In this extended sonification technique, the data stream controls parameters of a synthesis model, that in turn, controls the parameters of another synthesis model. (Scaletti)

To clarify these terms, imagine the shape of a graph that represents the variations of a fund on the stock-marked. The values on this graph can be used to control the pitch of a sound that is being played. This direct control is a 1st-order mapping of parameters. It will be an 2nd-order mapping, when the values of the shape control the amplitude of an oscillator, that in turn controls the pitch of another oscillator.
If only the second oscillator is used to produce the final sound output, the changes in the graph-values will result in changes in the spectrum of the sound output.

Realistic vs. Abstract Voices:

Realistic voices are those, that are either samples of real world sounds or convincing synthesised imitations of such sounds. Realistic voices have mnemonic qualities that can be of great value. Because these sounds are coming from real world events, the sources of those sounds can often quite easily be determined. This property of such sounds can be intentionally used if necessary.

Abstract voices are sounds, whose qualities are perceived without obvious associations to real-world sounds (Kramer2). Purely synthetic fantasy sounds are an example of such sounds. It is not possible to determine a realistic source for such sounds, that actually could exist.

Beacons:

Beacons are used as an extra aid in data representation techniques. They serve as absolute or relative references, to enable the user to navigate through data-sets or compare several data sets to eachother. They are auditory cues that indicate if a certain threshold is crossed, or they serve as a reference, like an auditory grid (Kramer2).
In a similar way (Gaver2) uses **sound holders**. These are objects that continuously emit sound, to enable a user to estimate his/her relative position, within an interface environment.

Auditory Icons:

Auditory Icons, further referred to as audicons, are the auditory equivalent of the visual Icon. They are also denoted to as caricatures of naturally occurring sounds.
The general, standard definition of an Icon is:

> A highly representational image, later combined with visual symbols.

Icons come in various forms. One of them is a representational icon, which is a simple picture of a familiar object of the real world. The auditory icons that are used in sound interfaces are, in design principle, very similar to this type of icon.
The limitation of a representational icon is that not all interface objects have a familiar or obvious pictorial representation, which means that they have no obvious real world representation. Nor does a well designed visual icon always have a good auditory equivalent. Take a document icon as an example. The visual icon is obvious, but what is the sound of paper? Only an action on such an object will produce sound, such as tearing up the paper, but this could also be interpreted as a document being destroyed/deleted.

(Gaver2) distinguishes three groups of auditory icons, on the basis of their functions for a single user.
1 Audicons providing confirmatory feedback to the user (redundant information).
2 Audicons providing information about ongoing processes and system states.
3 Audicons as aid in navigation within complex systems.

Earcons:
Another method of presenting auditory information are earcons. These are abstract, synthetic tones that can be used in structured combinations. In that form, sound messages are created that can represent parts of an interface. Earcons are composed of **motives**. Motives are short, rhythmic sequences of pitches, with variable intensity, timbre and register (Brewster et.al.). A motive can be used as a building block for larger groupings. The motives themselves and their compounded forms are then called earcons (Blattner2).

The interface aspects that can be represented by earcons, include:
messages, functions, states and labels (Blattner1).

By representing these aspects, the earcons provide information about:
computer objects; files, menu's and prompts
computer operations; editing, compiling and executing
interaction between objects and operation; for example editing a file.

(Blattner2) states that earcons are based on similarities between the auditory messages and abstract visual symbols. However, the relationship is mainly founded on the recollection of the earcon. In other words, the user has to recall what object or event is indicated by a particular earcon sequence.

Hearcons:
Although by the name, you might suspect that a hearcon is some kind of modified or enhanced earcon, a hearcon can in fact be everything from an audicon, earcon, to even speech output. The only restriction is that it continuously emits sound and that it is positioned somewhere in a virtual sound space, by making use of sound spatialization techniques. Hearcons can be used to represent objects, that are visually present on the screen, in the auditory domain. All the represented objects will then emit their own specific sound simultaneously. If the computer events are also included in the auditory representation, it may be that many sounds are present at each moment.

The characteristics of a hearcon are:
it's sound or tone-sequence, that represents the related interface object.
it's play-back volume
it's position-coordinates in space, in relation to a reference position
it's representative properties, e.g. the sound properties could give an indication of the size of the related object.

The possible user actions upon hearcons are selecting, moving and creating or deleting the hearcons. Similar manipulations can be performed on the visual objects that are present in a GUI-environment.

Filtears:
When the objects and events of a GUI are represented by sound, one could try to establish a tight interface by using the same basic sound material for objects or events that share aspects in common. The aspects that are different can then be represented by modifications of these basic sounds. These modifications should be noticeable, but not so drastically that they affect the identifyability of the original sound. To create such relationships, the sounds should be parameterized (Mynatt). This means that it has to be specified what modifications in the sound properties are introduced, according to the possible appearances of the interface objects. This looks somewhat similar to the sonification model used for representing data-sets. The sound parameters are attached to certain properties of the objects (to data properties in the sonification case), or to the way the objects relate to other objects.

E.g. a menu could have a certain sound attached to it. A menu-item could then be represented by a modified version of this basic sound that represents a menu.
For instance, the sound of the menu-item could have another pitch. In this way, it will be obvious that these two objects relate to one another, while simultaneously the difference between them is also made clear.
A second use of modifying the sound, is to provide information about the status of an object. When more properties of the basic sound are parameterized, even multi-levels of information can be conveyed.

All the possible changes in the sound parameters are systematically ordered and classified as filtears. In summary, the definition of the use of filtears is:

> ' convey added information, without distraction or loss of intelligibility or identifyability'.

By this definition is, in fact, every intended and noticeable change in a sound parameter that is used to convey information, the result of a filtear approach.

3 External experiments and system developments:

In the fields of data-presentation, sound-enhanced user interfaces and interface systems for blind people, several projects and system developments have been conducted. In this chapter some of the results of such projects and research are listed. To keep this paper in proportion, for more detailed descriptions I refer to the included reference literature.

3.1 Sonification projects:

3.1.1 The Sonification Tool kit (Kramer2):

G. Kramer developed a system for sonification research to test all kinds of sonification performances. It makes use of a technique called parameter nesting. It's general specifications are that the basic auditory variables; pitch, loudness and timbre, are divided into separate parameter levels. Each of these levels describes a certain property of the data that should be presented.

There were several practical problems encountered during this project. The first one was that parameters could overlap, which caused a loss of clarity. Secondly, sound parameters proved to be non-orthogonal. This leaded to not-intended changes in one set of parameters, when some parameters in another set were changed. E.g. a sharp attack also causes harmonics to be added to a sound. Attack-time thus interacts with brightness. A third problem that occurred was that increased polyphony could reduce the comprehensibility. In music perception it becomes increasingly difficult for most people, to follow each melodic line, when the polyphony in a piece increases. Single lines (monophonic music) on the contrary, are generally easier to follow. Contradictions on a cognitive level also occurred. Such occurrences will be discussed in chapter 5, section 5.3.

The problem of non-orthogonal sound parameters is a general problem in the use of sound and is therefore also of importance in the process of designing sounds.
The perceptual dependencies between different sound parameters have already been recognised for a long time. Therefore, it is not entirely clear, why this fact was ignored at the first instance, in the model that was used for the Sonification Toolkit.

3.1.2 Cerl/NSCE-project (Scaletti):

In this project he goal was to develop some prototypes for data sonification tools, that could be applied on a variety of time-dependant data streams.
Data sonification tools are techniques, that can be used for exploring, analysing and comparing data-sets by means of sound. Among others, auditory axes and grids were used and there was provided for several different ways of data comparisons.

Although no empirical evaluation with subjects was mentioned of in the article, it was noted that the use of instrumental timbres or musical scales might convey unintended cultural and historical meaning. The suggestion is to use this kind of sounds only when the symbolic meaning does not contradict or confuse with the indexical meaning. In a simplistic way this means that, for instance, a melodic pattern with rising tones cannot be used to represent a descending trend in the values of a data set.

For the design of earcons, this conclusion is not so relevant, because it is especially biased towards the use of audio in manipulating large data-sets. Still, it remains a fact that instrumental timbres and musical scales might be too tightly bound to purely musical applications. This should be taken into account when designing earcons. Earcons have to be created in such a way, that the musical associations are reduced to a minimum. Otherwise the user can be deverted from the pure information that the earcon should convey.

Projects addressing perceptual issues in Sonification Systems:

3.1.3 Streamer (Williams):

For this research the following question was raised:
"How are the components of an acoustic signal grouped together into perceptual objects, that the listener can interpret?"

The primarily concern of this research was investigating auditory streaming principles. These principles form a part of the auditory grouping processes. Auditory grouping is the perceptual process by which the listener separates out the information from an acoustic signal into individual meaningful sounds. The resulting related, single perceptual objects are denoted to as auditory streams. During the Streamer project, computational modelling and psycho-acoustic experimentation were combined, in an effort to trace out the interactions between frequency, time and amplitude aspects of the sound waves and the percept to which they lead. Especially those attributes that caused sounds to be grouped together, were a major topic of examination.

The experimental results demonstrated that the primitive grouping principles that apply, to segregate the components of the acoustic signal into perceptual structures, are highly complex-dependent. This means that no general rules could be found, that apply to all possible sounds. Also it was found, that the segregation performance of the subjects, depended on the previous experiences of the listener.

3.2 Auditory Icons:

3.2.1 Sound identification performance (Ballas):

These experiments were based on the question:
"What are the number of reasonable potential causes of a sound."
When this is known, the causal uncertainty can be calculated. The causal uncertainty, simultaneously reflects the number of possible, different causal attributes for a sound and the distribution of responses across these attributes.

The results showed a mean estimate that correlated significantly with the calculated causal uncertainty. Similar correlation was found for the identification time. This was the time that the user needed to come up with an answer. Further it was found, that the cognitive process of considering alternative causes, when presented with a sound, accesses a set of causes that is activated by acoustic retrieval cues. Memory aspects thus play an important role. When someone is confronted with a sound that he/ she has never heard before, most probably he/ she can think of many causes for that sound. On the contrary, when a sound is presented of which the cause is very familiar to the subject, he/ she will tend to recall this specific cause from memory.

Interesting is the influence that the context in which sounds were presented had on the identification task. Embedding the test sounds in a sound environment, seemed to have mostly negative effects on the accuracy, when compared to the test sounds being presented alone. With environments that didn't harmonise with the test sound, there was an obvious decrease in performance. Even consisting[1] environments didn't improve the identification tasks. The performance in such environments was either equal or worse, when compared to the case were the sounds were presented alone.

[1] Consisting, in this context, means that the sound is embedded in an environment that is strongly related to the test sound.

3.2.2 Comparing the identification performance of interface objects and events to that of their auditory equivalents (Leimann/ Schulze):

The main question addressed in these experiments was:
"Can sounds represent the objects and actions of the computer in the same intuitive way as icons do?"
To examine this, the research was split into three separate experiments. The visual- and audio cues were both presented and their resulting performance-rates were compared.

The question for the first experiment was:
"How well can the presented sounds be correctly identified?"
This experiment was somewhat similar to that of Ballas, described before. The mean result was that ± 58% of the sounds were correctly identified with the causal event they represented. No results of the performance of the visual cues were given in the consulted paper.

The second experiment dealt with the question:
"How well can the auditory and visual symbols be associated with the computer operation they represent?"
This resulted in a score of 36% for the sound representation and 40% for the visual icons.

In the third experiment, the effect of a learning period on the results of the assortment task was examined. The resulting mean scores after learning, were 58% for the sound representations and 64% in the visual icon case.

The general conclusion was that, although the visual icons have a better overall score, the results for the sound-representation are not far behind. However, the deviations in performance for different subjects, can become quite big when sounds are used. The relevance of this conclusion is that sounds could mean a fairly good substitute for the visual icons, as long as they are very carefully chosen and applied in a sensible way.

3.3 Earcons:

3.3.1 Investigation in the effectiveness of earcons (Brewster et.al.):

An experiment was designed to find out if structured sounds, such as earcons, were better than unstructured sounds for communicating information. With structured sounds are meant systematically designed sequences, while unstructured sounds refer more to sound bursts of synthetic or sampled sounds. Half of the unstructured sounds used for this experiment consisted of quite simple synthetic tones, like the system beeps of a computer. The other half were one-note sound burst, that had the same musical timbres as were used for some of the earcon pitch sequences.
Apart from comparing these two sound types, also an attempt was made, to find out how well subjects could identify earcons, when they are presented in different conditions. For this purpose the earcons were either presented individually or they were played together in sequences.

As an overall result, the experiments showed that musical earcons, consisting of musical timbres and containing rhythm information, were more effective than sound bursts, (like system beeps), or earcons composed of simple waveforms (like sin-, sawtooth- or square waves).

A second interesting outcome was that people could distinguish earcons individually. This on the contrary to recognising them on bases of hearing relative changes between the sequences. For example: a certain earcon sequence represents an object, that has state 'X'. Another earcon sequence, or a modified version of the previous one, can represent the same object, but with state 'Y'. If a change in state occurs, these two earcons can be played in succession. The differences between the sequences will be obvious to the listener. The sequences can be compared, because they are played close together in time. The experiments now showed, that if a single sequence is presented, that represents only state 'X' or 'Y', the user is still able to determine the state of the object. Thus a reference, embodied by the presence of the sequences that belong to the other state(s), is not necessarily. As a conclusion, this means that one, isolated earcon can be used to represent a specific object or event in an interface and provide absolute information about this event or object.

3.4 Other experiments:

3.4.1 Comparing the use of audicons, earcons and speech (Jones&Furner):

Some empirical comparisons between auditory icons, earcons and speech have been done by instructing subjects to:
a) Seek preferred sound-cue-types to given commands.
The resulting order in which the different auditory representations were preferred was:

1 speech, 2 earcons, 3 auditory icons.

b) Seek preferred commands to given sound-cue-types.
In this second experiment the preferred order was:

1 speech, 2 auditory icons, 3 earcons.

In the experiments also the effectiveness of the auditory representations was tested.
One of the conclusions was that auditory icons were 3th in preference, but 2nd in effectiveness.

In this research there was not accounted for learning curves. Also there was no special attention paid to investigate what the best and most adequate sound design for the audicon sounds could be. Therefore it is possible that, if other sounds had been used for the audicons, the results could be quite different. Changing the earcon sequences could have similar effects. The point is that there are no exact definitions for the final appearances of earcon sequences and audicons. These auditory cues can therefore have many forms, which makes an objective comparison of the usefulness of these two types of auditory representations very difficult. The only result that is widely accepted, is that the accuracy of speech output is always far better than that of either the audicons or earcons. The recognised problem with this kind of output are that it is slow and that not all possible events, happening in an interface, can be adequately translated into some kind of speech output. E.g. the appearance, or disappearance of windows or other objects are difficult to verbalise.

3.4.2 Audio cues for navigating through document structures (Portigal):

The experiment conducted by S. Portigal was primarily biased towards specific applications. The intended use of the sound cues that were tested, was to aid navigation through document structures. It was investigated if audio cues could be a useful means for navigation through the structure of a document, if compared to the visual cues. During the experiments the subjects were provided with:

the visual cues alone,
the auditory cues as a supplement to the visual cues, or
the auditory cues alone, as a replacement of the visual cues.

The results were that the combinational cue had no significant effect on the accuracy, as compared to the visual cues alone, but increased the working time needed. Providing only the audio cues resulted in an increase in the needed working time and a decrease in accuracy.

The conclusion for this experiment is that the sounds did not improve the performance of the subjects, when they were added to the visual interface. This does not mean that the sound-cues could not be used at all, but when they are used as only means for conveying information, a decrease in performance speed and accuracy, as compared to a visual presentation, will be inevitable.

3.5 Some computer systems that use sound for conveying information:

SoundGraph (Blattner, Mansur et al.(1985)): sonificates graph shapes. Some empirical tests indicated that a 3 [sec] sweep can give a rudimantair approximation of the shape of a curve. The sounds consisted of varying tone pitches .

SonicFinder(Gaver(1989)): desktop extension, using auditory icons. No empirical evaluations is known of .

SoundTrack (Edwards (1989)): an auditory word processor for the Macintosh. With this project, the visual interface should be made accessible for blind users by means of sound. The interaction had to be similar to the visual interaction. However, during the project constraints were applied to the interface design, to facilitate the use of sounds. In the end, the resulting interface was entirely adapted to provide visually impaired people with easy access to the implemented applications. The sound used for SoundTrack consisted of a combination of musical tones and synthetic speech. The timbre that was used for the musical motives, was a square wave .

ARKola (Gaver2 (1990)): simulation of background processes (also of other users) by means of everyday sounds. With 'every day sounds' are meant replicas of sounds, that can occur in everyones daily environment. No formal evaluation is known of.

AudioWindow (Ludwig, et al.(1990)): Digital Signal Processing (DSP) is used for shaping sounds and creating 3D-audio spaces. No formal evaluation is known of.

LogoMedia (DiGiano (1992)): auditory cues are used in a programming environment. They consist of various simple tones and sounds that have some recognisable characteristics. Some informal evaluation has been done .

Zeus (Brown, et al. (1992)): Sonification of sorting algorithms, by musical voices. No formal evaluation is known of.

AudioRooms (Edwards, et al. (1993)): A system that is still in design. It consists of icons that have spatial attributes attached to them. The techniques used for this purpose come from the Room project. Audio Rooms is a three dimensional presentations of the Rooms metaphor that was introduced at XEROX PARC. The aim is to create an intuitive desktop environment for non-sighted users. It is build around a low budged 56000 DSP-board , called the Beachtron, which can deliver sounds with 2D spatial cues, as well as distance cues. The used algorithms are based upon the technology that was used for the 'Convolvotron' at NASA-AMES. The 'Convolvotron' is a DSP-configuration that can be used for placing sound sources in a 3D-sound space (Wenzel et. al.).

Screenreaders: screenreaders are devices that translate the screen contents into a medium that a visually handicapped person can use. The most common examples are Braille and speech output. Screen readers came into being, among others, to enable visually impaired people to interact with a computer. Until recently most screen readers translated text lines into an alternative output and of such screen readers a lot can be found on the market. Only a few of them can also process the additional information of the interface-layout. An example of the later system is 'Outspoken', a screen reader for the Macintosh, developed at Berkeley Systems. Anyway, because the intended use of screen readers runs to far out of the scope of this paper, further information on the available screen readers and their details will not be listed here.

SPUI-B, a StereoPhonic User Interface for the Blind (Bölke&Gorny): the projects goal is to provide the advantages of window interfaces to blind users, without adapting the given GUIs. Every kind of sound can be chosen for the representation. These sounds are then called hearcons. The sounds are also spatialized by the use of headrelated stereophony. In this way, information about the spatial distribution of the interface objects will be provided and navigation can be facilitated. By the time of this writing, no empirical evaluation of the complete system has been reported of .

3.6 General trends & discussion:

It is obvious that much research has been done to enhance the usefulness of computer interfaces and the learning processes that come with the use of it. For the use of sound in interfaces, experiments have been conducted to investigate the effectiveness and the identification performance of audicons and earcons.

From the results, some guidelines for the creation of those auditory cues have been extracted , but no empirical data has been collected on precisely which sound could be used best to represent a specific interface object or event. This means that there is no all-embracing model, that defines exactly the characteristics of the sound properties, that should go with each interface object.

Most research, or developed systems, only tackle a small part of the problem. The cause of this fact is most probably the complexity of the problem . An auditory interface, that covers all aspects that are needed for a complete and workable replacement of the visual interface, is necessarily. For the creation of such a system, the combination of fundamental knowledge from many professions is indispensable. Important information can be found in the fields of psycho-acoustics, human psychology, music psychology , cognition, physiology, etc. Common interface design issues also have to be address. By combining the knowledge available in these fields, it should be possible to formulate rules that will lead to a set of sounds that is the most adequate to represent specific interface structures. Summarised: strict descriptions for every sound that is attached to a certain object, are necessarily, to be able to implement these sounds and create a sonic environment in which they will be embedded.

4 Psycho acoustical factors:

When using sound, the way in which people process and perceive sounds have to be taken in consideration. Among the important issues, for setting up a model to design useful sounds, are psycho-acoustical factors. For instance, sounds could indicate whether an object is selected or not. If the differences between the two sounds, that represent each state, are to small to be noticed by the user, confusion will be the result. Therefore it has to be known what the properties of our hearing system are. What aspects of sound can we hear and when will we be able to perceive differences in these aspects. Especially when more sounds can occur simultaneously, it is of importance to examine the effect that these sounds can have upon each other for our perception of those sounds. Such questions are dealt with in the field of psycho acoustic research. In this chapter, the psycho-acoustic factor's that are most relevant for the auditory interface design will be discussed.

4.1 Masking:

When two or more sounds are presented simultaneously, it is possible that one of them becomes inaudible or is reduced in it's perceived loudness, due to the presence of the other sound. This occurrence is called masking. Even when the sounds are not presented simultaneously, masking can still occur In this case it is denoted to as forward- or backward masking.

In an auditory interface design, these effects have to be taken into account. Especially of importance are the masking effects caused by broad band signals or noise. With such signals, there is an approximately linear relationship between masking and noise level (Houtsma). This means that, when the level of the masking noise signal is doubled, it's masking effect will also double.

Sounds appearing in a real world environment, are often broad-band and/or noisy signals. Because audicons generally consist of such sounds, the broad-band masking occurrences have to be considered when designing these sound cues.

The masking effects that tones can have upon each other is more of relevance to earcons. In general there is an increase in masking when pitches are close together in frequency. Also, tones mask other tones of higher frequency more effectively, than those of lower frequency (Houtsma).

4.2 Critical Bandwidth:

When broad band or noise-signals are used to convey information to the user, the critical bandwidth of the auditory system should be considered. This is necessarily, to be sure that the spectral changes that are made are also noticeable to the user. The critical bandwidth around a certain centre frequency is about 15% , (3 semitones), for frequencies above

± 500-600 [Hz] and 90 - 100 [Hz] for frequencies below this limen.

4.3 Just Noticable Differences (JND):

When using pitch and amplitude differences to convey information to the user, changes should at least exceed the just noticeable difference threshold.

For a certain frequency, the JND is about 1/30 of the critical bandwidth at that frequency. This will result in a JND of ± 3 [Hz] below 600 [Hz] and 0,005*Fc above this frequency (Fc is the centre frequency)(Houtsma). The JND value is not exactly defined. For example, according to (Zwicker), the value of the JND is about 0.007*Fc.

The JND for amplitude is about ± 0,4 [dB] in an completely ideal situation. However in practice the noticeable difference is a ± 10 % change in amplitude. This results in a difference in loudness level of 1 [dB]. Thus when loudness cues are used, the changes should at least exceed this 1 [dB] difference threshold to be noticed by the user (Temp).

4.4 Effect of duration on perceived amplitude (intensity):

If the duration of a sound becomes shorter than ±150 [ms], there will be a decrease in the perceived loudness of the sound, as compared to the same sound with a longer duration. This effect is a result of the logarithmical way in which we perceive the energy proportions of sounds, in relation to their durations. For a tenfold change in duration, the power must also change tenfold to keep the energy constant. In practice this means that by a doubling of the duration, the signal power must be changed by 3 [dB] (Yost). It can be assumed, that no essential sounds in the auditory interface will have a duration shorter than 10 [ms].

A sound of 150 [ms] will be perceived 11.8 [dB] louder, than the same sound with a duration of 10 [ms]. When this progression should be approximated by a linear slope, it has to be a slope of 0.08 to 0.09 [dB / ms]. Thus, if shortened sound-cues should be perceived equally loud as their longer equivalents, an amplitude correction of ± 0,09 [dB] for every [ms] that the sound cue becomes shorter, is necessarily (Duncan), (Yost),(Houtsma).

4.5 Loudness curves:

Another important issue in creating a balanced auditory display is the relativity of loudness perception. In general our auditory system is less sensitive for low and very high frequencies-ranges. Between 2 [KHz] and 6 [KHz] our sensory system is most sensitive, reaching a maximum in sensitivity around 4 [KHz]. However the variation in sensitivity over the frequency range, depends upon the overall intensity level of the presented sound.

ISOPHON-curves describe the levels at which all frequencies are perceived equally loud as a 1000 [Hz] (reference-) tone at a given intensity level.

To design a system where sounds are perceived equally loud, even when played over a large frequency range, these ISOPHON-curves can be used as a reference to calculate the compensation in amplitude that is necessarily to avoid large jumps in the perceived loudness when sounds are being played alternately in lower- or higher frequency ranges.

4.6 Discrimination of (changes in-) spectral shape:

For small bandwidths, a change from a flat spectrum to a spectral slope with an increase or decrease in amplitude of 0.38 [dB/octave] is noticeable (Versfeld).
When the bandwidth exceeds ± 6 [ST], the perceptible changes in spectral slope will be about 1 [dB/octave]. For narrow bands (< 3 [ST]), a stimulus change causes a perceived change in pitch. For broad bands (> 3 [ST]), a stimulus change causes a perceived change in 'sharpness'.

Another interesting fact is that the information that is being used by the auditory channel for discriminating spectral slopes, consists mainly of the edges of the noise bands, for bandwidths up to ± 9 [ST].

In practice the perceptibility of changes in spectral slope can be of importance when filtering is used as an information conveying technique. However the implemented changes in filter settings in the auditory interface are so drastically, that there is no doubt that they will be noticed by the user anyway.

5 Cognitive factors in the perception of sound. Association&Affect.

In the preceding chapter on psycho-acoustical factors, some properties of our auditory perception mechanism were emphasised. This is only one aspect of hearing mechanism. When it is known what we can hear, other questions arise, like what kind of 'feelings' can certain sounds elicit. When will sounds be interpreted as a warning signal, what properties can make sounds irritating or tiring and to what extent. Also what kind of causes do we naturally think of when we are confronted with certain sounds. E.g. with the sound of a driving car, we think of a car and by the sound of a big bang, people usually think of something exploding , or being hit.

In the total process of auditory perception there are three coarse distinctions that can be made (Williams):

sensation;

refers to immediate and basic experiences and responses, that are the result of isolated, simple stimuli.

perception;

is the interpretation of the sensations, giving them meaning and organisation.

cognition;

involves the acquisition, storage, retrieval and use of knowledge.

Psycho-acoustical factors belong to the sensational part of auditory perception and are often bound to physical properties of our hearing system. The purely psychological effects and/ or reactions that can be elicited by sound, are a somewhat different topic, that fits better to the last two groups; perception and cognition.

5.1 Sensation:

In sensation, psycho-acoustical factors (Chapter 4), play an important part. Additionally, human factors like the perceptual processing of noise-signals are also relevant topics in sensation (Rossing). Research on that topic has shown that, as far as psychological factors are concerned, performance isn't significantly affected by steady noise of less than 90 [dBA]. However, intermitted noise of this level can be disruptive. Some other interesting conclusions were:

Noise with strength in it's spectrum around 1000 - 2000 [Hz] is more disruptive than low pass noise.

Noise can affect the accuracy, but it has only limited effect on the quantity of jobs that are performed.

Noise can affect judgements of passed time.

Noise could cause feelings of fear and anxiety.

As a purely physical effect, sudden noise can cause people to prepare for defensive action against the noise source, revealed in muscular reflexes.

5.2 Perception:

In the chapter on existing systems and experiments, it was already pointed out that sounds are in general allocated to perceptional groups, or streams. This allocation process highly depends on the attributes of sounds that are perceived. Thus the process is not directly related to the physical attributes of the acoustic signal.

Therefore, the resulting percept may depend on attentional factors, on previous training or on familiarity with sounds that are similar to the ones that are presented.

The perception of streams is extensively treated in the field of gestalt psychology. Gestalt psychology is concerned with the human ability to recognise patterns and perceive configurations that appear in an environment. Gestalt in itself means an independent entity which has a definite shape or form. A short description of gestalt forming is that an organised pattern stands out from the ground field by it's contours and becomes a figure. It can only maintain it's structure by following certain principles. Some of these principles, that cause the forming of gestalts in our perception, will be discussed in this paragraph.

The knowledge in gestalt psychology is especially interesting for data-presentation applications. However, some principles apply more generally and are therefore interesting for auditory interface design as well. One of the fundamental principles in gestalt forming is **Pragnanz**. Pragnanz declares that a figure always becomes as regular, symmetrical, simple and stable as prevailing conditions permit. This is a rather open statement. A closer look at the processes, that determine the conditions is desirable. Processes promoting Pragnanz that are of interest for earcon and audicon design are:

Proximity: the closer two components are, the more likely they are to belong together. Some examples of primitive auditory grouping concepts for proximity are:

Temporal proximity;

when the time between two events becomes shorter, or when they even occur simultaneously, people tend to perceive these events as belonging together. In that case, one complete event is perceived, that consists of two or more smaller events, which form the building blocks.

Frequency proximity;

when the frequency contents of two events come closer together, for instance if the interval between two pitched sounds decreases, they are more likely to be interpreted as forming one continuos line. In this way, pitch sequences will form melodic patterns, instead of being perceived as a series of isolated tones.

Grouping by frequency proximity is also sensitive to the rate of presentation of tones. This results in apparent trade-off effects between the speed of presentation and the frequency separation. When the frequency difference between two groups of alternating tones increases, the rate of presentation needs to be slowed down, to avoid that the groups are being perceived as two independent sound sources.

Habit or familiarity: recognition of well known configurations, among possible sub-components leads to these sub-components being grouped together. For sound, the configuration of sub-components can be interpreted as the organisation of the harmonic structure. Some examples of primitive auditory grouping concepts for habit will then be:

Good harmonic ratio, which means a smooth spectrum that shows no interference from added non-harmonics.

Amplitude ratio inversely related to harmonic ratio, for instance a sawtooth wave has such a spectral envelope. It's timbre is quite pleasant and full.

With habits, the interpretation becomes a matter of recognising the meaning carried by the signal, not of only diagnosing its perceptual attributes. That is why the habitual processing of sound is actually more on a cognitive level, than on the perceptive level. A guideline for auditory display design can be extracted from the habit processes: Map data-properties to sound-aspects that are commonly associated with these properties, or to sounds that cause an affective response that suits well to these properties. E.g. low level percepts as increase in pitch&loudness tend to be naturally associated with increase in quality. Similar associational aspects will be more deeply discussed under 'cognition' (5.3).

Schemes: stored knowledge of familiar patterns may be used "top down" to assist in decoding the signal. E.g., speech recognition and the recognition of musical patterns and instrument timbres work in that way. If a particular scheme will be activated, depends on its level of familiarity and on how close this scheme matches the new auditory evidence, embodied in the presented sound. This dependence can lead to very interesting results. An acoustic signal that partially matches a very familiar pattern, may be recognised as this familiar event in preference of a pattern that it actually matches better, but that belongs to a more unusual event.

Belongingness: normally a component can form part of only one object at a time and its percept is relative to the rest of the figure-ground organisation to which it belongs. Conflicting relationships with other components create tensions in the field which must be resolved in order to achieve a stable state. The occurrence of 'belongingness' processes in our auditory perception can causes problems.

5.3 Cognition:

Knowledge from the field of cognition research may be used to maximise comprehensibility. For instance, one could use what is known about the metaphorical and affective associations, that can be elicited by sound :

Metaphorical association: the association of a change of a variable in the physical world with a, metaphorically related, change in an auditory variable. Some metaphorical associations, that are often experienced in practice (Kramer2):

louder = more;
This association might relate to the fact that, more and bigger objects make louder sounds, move more air and have a greater impact.
brighter = more;
A brighter sound is produced by adding more high harmonics and more energy (louder) to the upper partials of a sound.
faster = more;
When things move faster, more of a certain event occurs within a given time frame, while everything else stays the same.
higher pitch = more;
More vibrations per second result in a higher pitch. Also, adding up more to a pile makes it higher.
higher pitch = up;
High pitches or isolated frequencies are perceived as if the source is positioned higher in space. A second occurrence is the stretching of the neck, when people vocalising high pitched sounds. In music notation, higher pitches are also notated above the lower ones.
higher pitch = faster;
Faster vibrating objects, or faster running machines will produce a higher pitched sound.

Affective association: the association of 'feelings', that are aroused by changes in a presented sound, with 'feelings' about the data that is represented by this sound .
Such feelings are called the user's subjective affect. Thus the affect is induced by the emitted sound. This sound is in turn controlled by meaningful changes in the data.
E.g. as undesirable change take place, this event may cause a subtle sense of emotional discomfort to the user. This occurrence is then an affect. Examples of affective association are:

Ugliness:
a sound mutates from smooth to harsh, when high, non harmonic partials are added. This development will often be interpreted as the sound becoming more 'ugly'.
Richness to hollowness :
a sound mutates from a full spectrum to one without mid- spectrum components. People tend to associate such spectral developments with an environment or object that becomes increasingly hollow and bare.
Unsettling:
a pitched sound, created by two synchronised tone generators becomes less comfortable, when these generators are becoming detuned in relationship to each other. This effect can cause feelings of discomfort or anxiety.

Metaphorical and affective associations may interact or conflict with each other. For example: when more is associated with better, the brighter is more metaphor may coexists with the brighter is desirable affective association. On the contrary, if more is associated with undesirable, a conflict arises. The brighter sound will then still indicate that more of a specific event is happening. However, it should no longer elicit a pleasant affect, because the kind of events are not wanted by the user. As long as there is no obvious distinction possible in the feelings that are aroused by the two sound developments, the user can never be sure of the nature of the sound messages.

5.4 Functional aspects of sounds:

By intentionally using some of the mentioned perceptive and cognitive factors, it should be possible to design sounds that obtain specific reactions or elicit certain associations of the user. In this way sounds could be provided with informational functions, which, of course, would be useful for the design of an auditory-interface. The informational functions can be characterised by the following terms (Ballas):

Exclamation: get one's attention. Depending on the degree of urgency the spectral and temporal patterns of a sound can be varied. A relationship between the parameters of the auditory cue and the perceived urgency of a sound has been established by (Edworthly, Loxly and Dennis), leading to the next coarse classification:

Perceived urgency

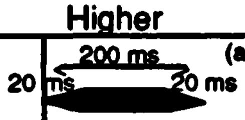
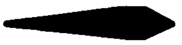
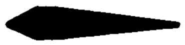
	Higher		Lower
Pulse		(approaching object)	(Object moving away)
Envelope			
Harmonic regularity	Random		Regular
Interpuls interval	Shorter		Longer
Burst			
Rhythm	Regular	Syncopated	Slowing
Average pitch	High		Low
Pitch range	Large	Small	Moderate
Pitch contour	Random(atonal)		Down/Up

fig. 5.1. Basically from (Ballas, J.A. Delivery of information through sound).

For every evènt that should be audificated, the level of urgency can be determined. This could mean that the use of a certain spectral slope is excluded, if the level of urgency demands that.

Deixis: pointing to some event or entity that is related to the (computer-) operation. For the understanding of deictic signals, contextual information, such as position and referenced object, is necessarily. A most widespread example of a deictic signal is the well-known computer beep, which makes no sense without some additional contextual information.

Simile: likening sound properties or parameters to the properties of the other domain that they represent. Associative behaviour (§5.3) is related to this kind of function.

Metaphor: the sound and the object that the sound represents, are interpreted as being the same thing. Metaphors that have become common in use and language are called dead-metaphors. Examples of dead metaphors are: the telephone (ringing) or the doorbell. In these examples, the object and it's sound are fused into one percept. Learning is thereby be an inevitable aspect. This, because dead metaphors will only emanate from the fact that people are confronted so often with certain sounds and the related objects, that they automatically start to interpret these aspects as being one and the same thing.

Onomatopoeia: representing events, by imitating the sound that is typically produced by the occurrence of these events. The communicative function of the event's typical sound should therefore be retained. Factors that are involved in the interpretation of sounds :

a Acoustical properties;
the purely physical aspects of the produced sound waves.

b perceptual interrelations;
the perceptual processing of the properties of the presented sound (§ 5.2).

c cognitive expectancies;
all the aspects that have to do with the cognitive interpretations that are evoked by the perceived sounds (§ 5.3).

d ecological frequencies;
the influence that the (sound-) environment, in which the sound events occur, has on the interpretation of those sounds.

Cognitive expectancies appear in two types, which are:

Expectancies about the usual cause, that is associated with a sound. This phenomenon was examined in the experiments, that were described in chapter 3, § 3.2.1.
The results of this experiment were expressed such expectancies in degrees of causal uncertainty.

Expectancies about the typical sound that is normally associated with a certain cause. This type encapsulates the mental concept that people have of a sound. The sound sets that were used for the experiments described in § 3.2.2, were actually based upon such expectancies. For the creation of those sets, many people were asked to come up with sounds that, according to their meaning, fitted best to a certain object or event.

6 Common human factors in the auditory interface design:

Apart from the factors, belonging to fields like psycho-acoustics and cognition, there are some additional human factors to be considered. Topics relating to the fact that we are actually working with an interface, are also of importance. These are factors that partly apply to Human Computer Interface (HCI-) design in general. Therefore, in this chapter, some general issues in designing interfaces are discussed, together with aspects like; the timing of movements that users perform and the influence of memory aspects in using an interface.

In terms of common HCI-design terminology, audicons can be seen as a **hot medium**. This term means human-computer communication, whereby the meaning is delivered in the message itself. Earcons can in the contrary be considered a **cold medium**, which is human-computer communication, whereby the meaning of the message is furnished by the receiver. In practice these very strict distinctions are difficult to be made, due to all kinds of mixed-forms of audicon & earcon properties.
E.g. if the sound of an audicon also encapsulates tone-sequences of some kind, earcon aspects and properties are addressed as well.

6.1 General interface design heuristics (Thimb):

In relation to some the common HCI-design terminology, it is useful to take a look at some HCI-design heuristics and see if they can also be of use for the auditory interface design. The process of designing can be separated into three distinct steps which are: **Generalise, specialise and explain.**

1 **Generalise**: design features that share aspects in common, should be spotted and replaced with one common feature. The following listing provides an overview of the desired properties for the features. The possible applications of these heuristics for an auditory interface are given in addition. The features should be:

Consistent.

Orthogonal: the component functions should be independent.

Appropriate: only essential functions should be provided.

Parsimonious: each function should have a single form.

Transparent: the way generalisation works, should not be visible (audible) to the user.

Powerful: each function should do as much as possible.

Open-ended: there should be freedom to combine functions and implement new functions.

Complete: the generalisation should do the complete job.

In interface design it is necessarily to make some generalised commitments in advance. In these commitments is put down, who will use the system and what the system will be used for. A few kinds of commitment are possible, but the one that is of importance for this paper is: Pré-commitment. It is known in advance who uses the system and what the system will be used for. All requirements that the system should meet can already put down in advance. This type of commitment can lead to an eager evaluation system.
In an eager system, events are processed at once and very complete (in batches).
It is only possible to set up a system in such a way, because of the pre-commitment. When the intended use of a system is not exactly known in advance, that can only become clear, during the development. In that case the system should be adapted along the way, leading to a less accurate system, that functions more step by step and order-dependent.

2 Specialise: use the outcome of a commitment to target a generalised idea towards a more particular goal. It is wise to postpone the specialisation as long as possible.
In that way it will remain possible to make changes, during the development of a system.

3 Explain: extract the general principles on which the system design is based. The system must be closed under this explanation.

6.2 Timing:

Apart from the general design principles, human factors, like physical- and timing aspects have to be considered. An important combination of both aspects is the influence of physical factors on the amount of time that user movements with the cursor consume. In HCI-design, some laws for calculating the duration of certain operations have been derived. These are:

Fitt's laws: [time to move your hand] = log (target size / distance)
and
Hick's law: the user's decision time is proportional to: log(number of choices)
or even longer.

It is important to examine if similar laws apply, if not an ordinary interface is used, but an alternative interface that gives visually disabled people access to the computer.

6.3 Memory:

A last item to be mentioned in relation to interface design is the influence of memory on the processing information. In general most models of our memory make a distinction between short term and long term memory. If more directed towards the processing of auditory information, a similar distinction is possible.

Echoic storage: during this amount of time that a sound is stored, processing of subjective auditory duration and auditory integration on it, is enabled. (Portigal-> Lachman). This part of memoric storage is related to the effect of duration on perceived amplitude (§4.4) and is called the **short auditory store** by (Crowder-> Cowan).
The time duration of echoic storage is about 130-250 [ms]. It contrasts with the long auditory store, which covers ± 2 - 10 [sec].

7 Graphical User Interfaces and visually disabled users.

Until this far, many important factors that are related to sound have been discussed. Several times a glance has been given on the intentional use of sound. The sound design process that is discussed in this paper, is part of a project for making Graphical User Interfaces (GUI's) accessible for blind people (Poll1 et. al.).
The screen contents of a GUI consists of graphical objects, that represent functional or informational aspects that are present in the computer system. An example are graphical objects (Icons §2.2), that represent the available application programs on the current system. The most common way in which users can interact with a GUI, is by using a pointing device, called the mouse. This mouse enables the user to move a screen pointer over the screen contents. With this screen pointer, the objects that are present on the screen can be manipulated. The manipulations that are possible, like selecting, activating and dragging are listed and explained in chapter 11.

Blind people are not able to make use of the information that is visually presented on a computer screen. When only text is displayed, the blind user can make an appeal to one of the currently available systems, that translate the textual information into a sensory domain that visually disabled user can perceive. Examples of such systems are screen readers, that translate the text, displayed by a computer, into Braille or synthetic speech output. For GUI's, alternative solutions have to be found, because text is only a part of the total amount of visual information that is provided by the computer. The current project is concerned with finding such a solution.
A restriction for this project is, that the visual layout of the screen contents is not altered. Also the way a GUI is generally used, should stay as unaffected as possible.

In the alternative interface, that should provide visually disabled people access to a GUI, the basic interaction principles should be maintained. This means that the mouse is retained as input device, although in a slightly adapted form. The position on the screen, that is controlled by the physical position of the mouse, has been changed from a relative, to an absolute position. This has been accomplished by using a digitizer platform with upstanding boundaries, that represents the screen dimensions.
A wireless mouse can be moved across this platform and it's positions on the surface match to the positions on the screen. The input system, that consists of the platform and mouse is called the **SoundTablet** (Poll1 et. al., Poll2 et. al.)

The SoundTablet is only one part of the complete system, that is needed to make a GUI accessible for blind people. In outline, the rest of the system consists of:

The **target computer**: a standard personal computer, running Microsoft Windows, without any constraints or adaptations on what is visually being presented on the screen.

A **hardware bridge** that reads, recognises and translates the current screen contents. This bridge is implemented on a separate computer and consists of:

Video Interface and Signal Analysis (VISA) hardware. This hardware converts the video signal of the screen onto a bit-map, that will be used by the software modules, that form the subsequent part of the system.

VISA software. The first part of this software consists of extended *Optical Character Recognition* (OCR) software. The OCR-software recognises the interface-elements in the raw bit-map data, that is provided by the VISA hardware. In connection to the introduction of GUI's, the extended OCR recognises both text and 35 visual elements (objects), other than text. A *multitasking kernel* routes the data, that is produced by the OCR, to a *screen construct/update module*. This module analyses the OCR data and derives GUI objects from the recognised visual elements.

Finally, the *Current Screen Model* (CSM) is updated on basis of the constructed objects and according to optional user actions. Among others, the multitasking kernel keeps track of the user actions and provides them to the CSM and the screen construct/update module.
Fig 7.1 illustrates the functionality of the hardware bridge within the complete alternative interface system.

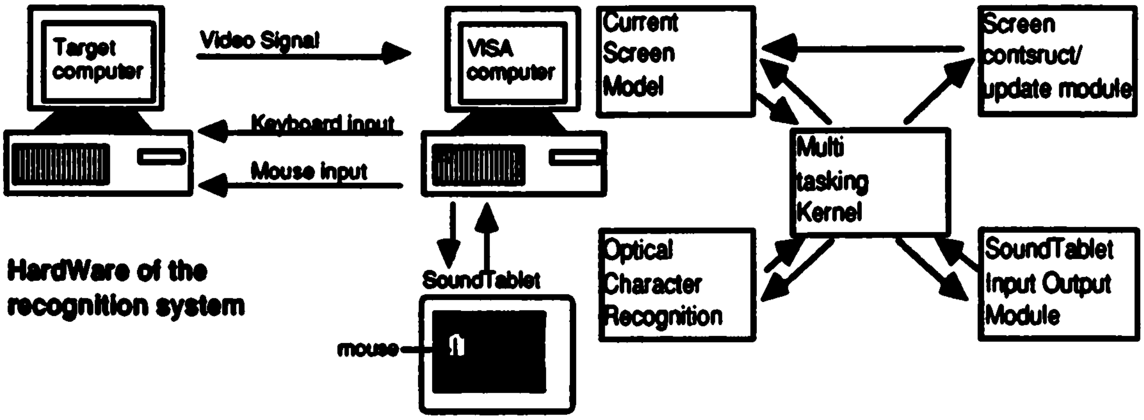

Target computer
Video Signal
VISA computer
Keyboard input
Mouse input
SoundTablet
mouse
HardWare of the recognition system
Current Screen Model
Screen contsruct/ update module
Multi tasking Kernel
Optical Character Recognition
SoundTablet Input Output Module

Fig. 7.1

The system displayed in figure 7.1, will take care of the translation and recognition of the screen contents of a GUI and keeps track of the user actions and manipulations, that are performed with the mouse. Still, a system is needed that transfers the captured GUI-information to the visually disabled user in a form that this user can perceive. For this project it is chosen to do this by using speech and non-speech audio . In spite of the reduction in bandwidth, as compared to the visual channel, serious attempts are made to convert as much as possible of the information of the user interface into the auditory domain. The choice where to use non-speech or speech output, or even both kinds of sound output simultaneously, depends on the kind of interface aspect that should be represented. Some aspects, like text-labels, are very suitable to be presented by speech output, while others, like typical computer events, have no obvious textual representation. Such events can then be revealed to the user by playing non-speech sound cues. The additional hardware that is needed to provide the sound output consists of:

A **sound card** for the PC, that provides the sampled and/ or synthesised speech output.

Some hardware, that provides the sampled and/ or synthesized non-speech sound output. For the current system set-up a MIDI **Sampler/Synthesizer** is being used for this purpose.

Software, that is needed to drive the sound generating hardware, has also been developed (Poll3).

Recapitulation:
The system translates the visual aspects of a graphical user interface into the auditory domain. In this process, the original lay-out of the GUI is maintained, as well as the interaction methods that are typical to a GUI. In addition, an early attempt is made to aid navigation and orientation within the auditory space that is created. A method has been developed to provide the user with information about the position of the pointer and the relative position of the objects, within the interface. This method is called the Auditory Guidance and will be discussed on more detail in §17.2.4.

8 Psycho acoustical, cognitive and HCI factors in auditory interface design

The topics that have been discussed so far are valuable for the design of an auditory interface. They can either serve as a reference for setting up requirements that the sounds should meet, or they can aid in structuring the broad spectrum of all the aspects that are related to the use of sound in an auditory interface. Of some of the discussed terms, or topics, their meaning for auditory interface design is listed.

Which approaches, that are indicated by the terms of §2.2, are most often used?
For replacing or enhancing the visual user interface, audicon and earcon approaches are most widely used, while the application of hearcons is a more recent development. Unfortunately, when a GUI is represented in practice by hearcons, some problems occur. The auditory senses cannot process the amount of information, that is provided by all the hearcons simultaneously. Searching adequate solutions to this problem is still one of the main topics for inquiry (Bölke&Gorny).

What meaning in sound perception has 'belongingness', as part of cognition processes ?
As far as cognition is concerned, the occurrence of 'belongingness' processes (§5.3) in our auditory perception can causes problems for auditory displays. A percept that is attached to a particular data attribute can be hidden or distorted by the surrounding context. Also, when data items that should be contrasted are attributed to different perceptual objects or streams, that seem to have no relationship to eachother, these items will be perceived as isolated events, instead of contrasted items.

How can the functional aspects of sounds as discussed in §5.4, be applied to auditory interface design?
Exclamation:
Exclamation properties can be used for representing the appearance of dialogue boxes, or indicating when objects are hit with the mouse pointer. For such events the level of urgency can be determined. The sound parameters can then be tuned to the required properties that are attached to each urgency level.

Deixis:
A representative of a deictic function in an auditory interface is the indication that the mouse pointer is in a 'busy-state'. Within Microsoft Windows, this state will be indicated by the shape of the pointer that turns into a hour-glass.

Simile:
This functional aspect of sound can also be applied to an auditory interface.
The properties of the auditory alternatives for the visual objects and events, should then be harmonised to the properties that these visual items embody.

Metaphor:
Metaphoric behaviour, represents the ideal application of an earcon. When designing earcons, one should try to obtain earcons that are dead metaphors as end result.

Onomatopoeia:
The function that onomatopoeia represents, is in fact the underlying basis for Auditory Icon design.

How can general HCI interface design principles be of use for an auditory interface?
For common user interface designs, a generalised commitment is put down in advance (§6.1). The type of commitment that applies to the design of an auditory interface design, is pré-commitment. What such a commitment implies, is explained in §6.1. Apart from a commitments, also a specialisation is done (§6.1).

Commitment and specialisation can also be applied to an auditory interface. For such an interface, the pré-commitment consists of the facts that:

- the users will be visually disabled people,
- a GUI should be made accessible for these people, so that they can use the computer in an office environment,
- providing this access, should be done by using speech and non-speech audio.

The subsequent specialisation might be embodied by the fact that a mouse tablet (Sound Tablet) is used, to maintain compatibility with the interaction mechanisms of the actual GUI. Specialisation is of less importance in providing admission to the structure of a GUI, since there will be no new user interface or application designed from scratch. An already existing interface has to be translated into the auditory domain, but the specialisation of the features of the GUI environment, has already been done during the design of this graphical user interface.

Explanation is a third step in a structured interface design process.
Good explanation is also an important issue in the auditory interface design, especially when earcons are used. The users cannot know in advance, what all the auditory cues mean or represent. Therefore accurate instruction and explanation to the user is necessarily.

What is the significance of timing issues for the use of an auditory interface (§6.1)?
During the SoundTrack project, (Edwards2) found, that the mean moving-time between objects was 0.73 [sec]. However, the SoundTrack interface representation was especially adapted for blind user and therefore not representative for GUI's in general.

The results from some experiments on the usefulness of the auditory guidance in a GUI-like environment (Poll4), indicated that the mean time it took to move between objects, was 0.69 [sec], with a standard deviation of 19. The objects were spaced at a 30 [pixel] distance apart. This means that 97,5% of the movements were slower than 0,01 [sec/pixel]. So 100 [pix/ sec] can serve as a reference value for the maximum speed. The practical use of this reference value will be more deeply discussed in §16.2.4 and in appendixA §3.5.

The processing of speed information is of importance for sounds, that aid navigation, or that are otherwise directly related to movements with the mouse pointer. An example is dragging of objects. Apart from Doppler effects that could be introduced, it is necessarily to find an optimum for the update rates of the guiding sounds. The pulses should be fast enough to keep track of the pointer position, without distracting or irritating the user. These undesirable effects can be caused by too turbulent pulse rates, or too much annoying variations in the update rate. If 23 [ms/pix] is used as an indication of the average speed and the maximum acceptable difference between the actual position and the 'sound-playback-position' is 'X' [pixels], then the interval time between the sound pulses should be at most, X.23 = Y [ms] .

9 Guidelines for the creation of sound sets.

Various sound aspects have been treated in the preceding chapters. When the knowledge of these previous chapters is combined, guidelines that can serve as a reference, can be put up for the actual creation of the sounds, to be used in an auditory interface. Although most factors, that are listed here, have already been previously discussed in more detail, an effort is made to retain a complete as possible overview. This overview will be followed by a stepwise summary of how the process of making sounds for an interface can be tackled (chapter 10).

9.1 Guidelines related to psycho-acoustical factors.

If noise signals are used for conveying information, make sure that the distinctions in frequency contents between different noise signals are noticeable. An obvious way to ensure this, is by making use of sounds of which their main frequency ranges are at least spaced one critical bandwidth apart.

It is difficult to draw up some objective criteria for the perceptibility of changes in the spectra of complex timbres. However, the findings of (Versfeld) (§4.6) can serve as a reference, when changes in the spectral slopes of noise bands (filtering), are being used to convey information.

Information about the interface could be embedded in differences in amplitude and / or pitch (for clearly pitched sounds). If such a mechanism is used , it should be ensured, that these differences are far over the Just Noticeable Difference thresholds (§4.3), so that these changes cannot be missed.

To create an auditory interface that is well balanced in amplitude, use a loudness compensation curve. What reference curve should be chosen, depends on the mean amplitude that dominates within the used amplitude range.

To avoid masking, use broad-band and/ or noise-signals only at a low amplitude level. This level should be compared to the level of other sounds, that can occur simultaneously. A low amplitude for noisy sounds is also recommended for other reasons then masking. Long lasting, high amplitudes could cause distraction and listeners fatigue. Sounds that can occur simultaneously or at close distance in time[1] should be judged by the masking effects, that they can have upon eachother. This could be done by analysing the sounds with an analysis program, that contains psycho-acoustical models of our auditory system. Many programs or routines, used for data-reduction based on psycho-acoustical models, already make use of such models.

[1] Forward masking can occur up to 20 a 30 [ms] separation in time, backward masking up to 10 [ms], although this last kind of masking is in general less effective. (Houtsma)

9.2 Guidelines related to sensational factors.

Avoid the disruptive effects that noise signals can induce. To do this, use steady noise and keep the noise amplitude level low.

The amplitude range in which all sound events will occur should not be to wide and the overall amplitude should not be to loud. This to avoid listeners fatigue and distraction, caused by continuously emitting sound at high intensities. Large jumps in amplitude, that could distract or startle the user should also be avoided. Guidelines resulting from research done on earcons, can be more generally applied as well. In these guidelines it is suggested, to keep the intensity levels close together. The suggested ranges are:

max. 20 [dB] and min. 10 [dB] above hearing threshold.

9.3 Guidelines related to perceptional factors.

Tone sequences, that are related to an object or event, can be treated as streams. Therefore, the principles of gestalt psychology (§5.2), also apply to earcons. The importance for the practical use of these principles varies, it depends on facts like:

can two or more earcons sound simultaneously and
exactly which earcons will be played then.

E.g. suppose that two simultaneously playing earcons make use of the same timbre. The proximity- and scheme principles will then be of importance, to be able to ensure that the earcons can still be distinguished from one another. In general, one could apply principles of habit, scheme and proximity in such a way that:

Earcons that represent related objects/ events will also be perceived as related.
Earcons will form compact structures, that can easily be recognised/ distinguished among other earcons. Compact structures could be obtained for instance by keeping the intervals within one sequence small.

9.4 Guidelines related to cognitive factors.

Metaphorical and associative factors can directly be applied in the way they are listed in chapter 5, according to the intended association that an object or event should elicit.

For making use of the informational functions that can be attached to sounds (§5.4), the specific interface function that an object or event represents, has to be determined at first. Then, the sound properties that fit to this functional element can be selected. Some examples of the use of the functional aspects of sounds are given in chapter 8.

9.5 Guidelines related to common human factors.

From the requirements that interface features should meet (§6.1) and that are part of the process of generalizing an interface, some guidelines can be extracted for the auditory interface design. The feature requirements, can be interpreted for an auditory interface as; Features should be:

Consistent:
To obtain consistency in auditory interface design, try to use similar sounds for all objects / events, that share similar properties and functional aspects. In that way, the underlying meaning of the sounds can be more easily recognised and the total number of different sounds can be reduced.

Orthogonal:
Because of the interactions between different sound-parameters when they are processed by our auditory system, real orthogonality in sound-parameters is not possible. For instance changes in loudness also have their effect on pitch perception.

Appropriate:
Try to avoid that events or objects of minor importance are assigned to sounds, that have a perceptually equal weight as the sounds of more essential objects or events. Otherwise there will be an unbalance in the interpretation of the importance of the presented objects and events. (See also the cognitive level of urgency §5.4).

Transparent:
In the ideal situation, the user should not have to know anything of the hierarchy and classification of the sound objects. The sound presentations should 'speak' for themselves.

Powerful:
This can be interpreted as: one earcon or auditory icon should cover as much as possible of the object's or event's properties. To accomplish this, it is better to translate every object property into a sound parameter, than to assign a completely different sound to each object property.

Open ended:
In an auditory representation, several earcons consisting of short motives can be combined into larger earcons. The resulting sequence will then contain more information about the objects or the events that are represented. Such techniques are often used in earcon design for applications with a data-representation purpose.

Complete:
For the design of an alternative representation of the visual windows interface, this last requirement is of less importance. During the project, no new interface structures are designed. The only goal is to make the already existing ones accessible to the blind users. Therefore, the alternative representation includes all the good and bad aspects of the windows interface, that are already there. It is of no interest, whether the visual presentation of the window environment meets all the listed qualifications, or not. (Unfortunately, it doesn't meet all the criteria, but that's up to Microsoft Windows!).

In an auditory interface the aspects of auditory memory are of importance, especially the echoic storage:
According to the amount of time that audio information is stored and processed in short-auditory-memory, it is useful to keep one-shot sounds within a duration of 250 [ms] (§6.3). For more continuous sounds, similar constraints can be applied. One should try to keep all the essential information within the onset phase of the sound. This essential information is the informational function, that the sound should transfer to the user. Changes that occur when the sound already plays, should also be represented by an auditory cue, that has the short duration of about 250 [ms].
Keeping the durations of the sounds or changes short, also makes practical sense in reducing the auditory load on the user and avoid slowing down the performance to much.

9.6 Guidelines for the creation of earcons.

Some overall guidelines for the creation of earcons have been extracted from results of several investigations into the effectiveness of earcons (Blattner2, Brewster et.al.):

1 Use musical instrument timbres, with at best multiple harmonics.

2 Do not use pitch information on it's own, thus as only means for conveying the interface information. Suggested ranges for pitch are: maximum 5 [KHz], and minimum 125 to 150 [Hz]. Changes in pitch are most effective when they are used in combination with changes in the rhythm.

3 When register is used as an independent parameter, large differences should be used (several octaves).

4 Rhythmical changes must be as different as possible. Different numbers of notes in each pattern seems to be an efficient way to accomplish this. The shortest note length should be about 0.125 [sec]. When it is assumed that the shortest used note value can be a 16th, the tempo should not be higher than 125 bpm. Taking into account the variations in time of the transient phases of the sounds, a maximum tempo indication of about 120 bpm should be sufficient.

5 Intensity levels must be kept close together. Suggested ranges are: max. 20 [dB] and min. 10 [dB] above the threshold, that is present due to masking back ground noise. Thus a sensible amplitude range reaches from 10 [dB] until 30 [dB] above hearing threshold.

6 When earcons are played in succession, a gap in between them of min 0.1 [sec] should be maintained.

10 The design process in a stepwise overview :

With the collected knowledge and the guidelines, the complete process of designing and creating sound sets for an auditory interface can be outlined into the following steps:

1 First a GUI interface, that should be provided with an auditory equivalent, is chosen. Then, all the objects and events that have to be provided with non-speech audio should be extracted and categorised, including the user actions that have to be audificated (C11).

2 Objects&events have to be listed conveniently, together with all their possible states, their functional aspects, the possible user actions upon them and optionally a short description (C11).

3 According to these attached properties the objects and events can be divided into families that share aspects in common (C11).

4 Depending on the objects' and events' properties and the fact if they should sound continuously, as a one-shot or as a pulsating signal, sound-properties that most suitably represent the aspects of each family can be drawn up, according to the guidelines for the creation of sound sets (C12).

5 Next, lists of potential sounds, that satisfy the desired conditions can be created for each family (C12).

6 Then the sounds can be coupled / assigned to the objects and events (C11).

7 Of the proceeded sound sets, the amplitude and the frequency regions that the sounds occupy, have to be calibrated. Sounds that can potentially mask eachother, have to be analysed on their masking effects and judged on their usefulness. Possibly, adaptations according to the results of the analysis are necessary .

8 Final adjustments have to be made, according to the results of the masking-analysis and eventual other problems that are encountered. E.g. such a problem, that is not directly related to calibration or masking-effects, could consist of difficulties in the identification of timbres. Since there are no objective criteria for the definition of timbre, timbres that are easily recognised when presented alone, might turn out to be difficult to be distinguished from other timbres, when presented in a complete sound set.

The letter/ number-combination between parentheses, indicates the chapter of this paper, in which the mentioned steps are integrated. E.g. C11 means, that the preceding step can be found in chapter 11

11 Listing and classification of the interface objects and events.

According to the stepwise approach, suggested in chapter 10, the first step of the process is sorting out the objects that have to be provided with the non-speech audio and categorising them, according to their properties. Thus to start with, the objects and events are sorted, subdivided into families and put together in a list. (Step 1,2&3). A complete listing and extensive description of all the interface objects and events, can be found in (Gerrits et. al.).

11.1 Informational and physical level.

A classification according to the function of the interface items, within the interface should be made. To accomplish this, a coarse distinction in an informational level of the user interface and a physical level is necessarily. The informational level contains attributes like the identification, the state and the events of the objects.
The physical level describes the border, surface and position information of objects.
In figure 11.1, these levels are displayed in a diagram:

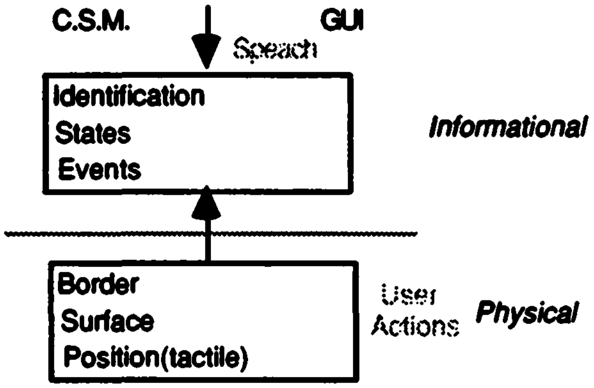


Fig. 11.1

For the current project is decided, that the attributes of the informational level are always presented by speech output, while the physical attributes are represented by non-speech audio. This doesn't exclude the use of non-speech audio for the informational level, but in that case it will serve as an addition to the speech output.

11.2 Objects&events list.

Table 11.1 displays the resulting object list. An explanation of the properties of the interface objects will follow this list. The parentheses around some of the listed objects or events, indicate that these items are not yet implemented in the current system.
More on that will be discussed in chapter 13. In connection to the available space, some of the object states have been abbreviated. The complete names of these objects are:
def is default, PD means Pull-Down, Hormenu-list is horizontal menu list and sel is an abbreviation of selectable.

Object name:	Event:	User action:	Object state:
Fam A:			
Button	hit, above	selecting	def, non-sel, selectable
Check-box	hit, above	selecting	un-marked, def-unmarked, marked, def-marked non-sel
Radio Button	hit, above	selecting	off, on, def-on, non-sel

Fam B:			
Edit-Control	above	-	active, non-active
Edit-line	hit, above	selecting	selected
Text-line	hit, above	selecting	selected
List-box	hit, above	-	-
List-as-PD	appear, disappear	selecting	-
(list)	-	(scrolling)	-
Fam C:			
Menu-list	appear, disappear	selecting	-
(Hormenu-list)	(above)	-	-
Menu-Item[1]	hit, above	selecting	non-highlighted def, non-sel.
PD-edit	hit, above	selecting	non-highlighted def
Icon	hit, above	selecting,dragging, (activating)	non-highlighted def
Fam D:			
Dialogue-box	appear, disappear, hit, above	selecting[2], dragging	active, non-active
Window	appear, disappear, hit, above	selecting, dragging, (mini-/maximise, resize)	active, non-active
Title Bar	above	selecting, dragging	-
Fam E:			
(Group Box	inside		-)
Graphic	hit, above		-
Pointer	-		busy
FamF(events):			
Hit1	not applicable	-	hit: in general
Hit2	not applicable	-	hit: window/ dialogue-box
Select1	not applicable	-	select with cursor
Select2	not applicable	-	select by default (return)
Drag	not applicable	-	continuous
(activate	not applicable	-	)
(minimize	not applicable	-	)
(maximize	not applicable	-	)
(resize	not applicable	-	continuous)
FamG:			
AuditoryGuidance			continuous

Some footnotes to this classification:

[1] Group-borders, that are sometimes present in menu-lists, form a visual separation between groups of related menu-items. In fact they are also menu-items, however they are empty (have no label) and are non-selectable. Because they contain no textual information, the group borders can be represented by playing only the menu-item sound. The speech-output that indicates the informational aspect of the object can be omitted, because only the physical level exists.

[2] In the case of a non-modal dialogue box, it is possible that the dialogue box is inactive. Non-modal means a dialogue-box, that doesn't have to be settled before the user can continue with other tasks. If it is non-active, it has to be activated again by selecting, if the user wants to continue it's settlements related to the dialogue box.

11.2 Description of the interface terms.

In this paragraph the exact meaning of the interface terms will be discussed.

11.2.1 Explanation of event types:

Events will occur when the user is navigating through the interface. They indicate dynamic or fixed system properties. Most events will be triggered as a result of user actions. On the physical level can be found:

1 Hit: the border of an object is hit with the pointer device.
2 Above: the pointer is moving within, or fixed above an object's region.
3 Inside: the pointer is moving within an enclosed area, like a group-box.
An enclosed space is visually represented by an outlining, or the area can have another back-ground colour than its surroundings.

On the informational level there are:
4 Appear: an object appears / pops-up. Appearance is caused by a direct or indirect user action or an automatic system or program routine.
5 Disappear: an object disappears. Again, direct or indirect user action, or automatic system or program routines will caused this event.
6 State change: an object changes in it's current state (e.g. default-> non-default).

11.2.2 Explanation of user-action types:

User actions are active, conscious actions, explicitly performed by the user and not automatically resulting from pointer positions and / or changing system or program states. User events can also trigger some of the other event types. E.g. activating could cause a window to appear. The user action on the informational level are:

a Selecting: selecting the object where the pointer is currently above. A result can be that the object gets highlighted and/ or becomes the currently active object.
b Activating: some objects can be activated after they are selected. Selecting and activating are often performed in one run, by double clicking. Activating can also cause objects to appear or disappear.
c Dragging: some selected objects can be dragged. To do this, the mouse button shouldn't be released after selecting the object. As long as the button is pushed down, the selected object can be moved.

User actions that contain aspects of both the physical and informational level are:
d Minimising: this action is only of important for window-objects. Minimizing reduces the size of the window to the size of a usual icon.
e Maximising: this action causes a window to become the size, that is needed to display all objects that it contains. This could mean that the whole screen size will be filled with this window, if the amount of space that is needed, demands this.
f Resizing: the user can resize a window horizontally and vertically to his needs.
This can be done by selecting+holding the resize corners of a window. This corner can then be dragged until the appropriate size is obtained.

11.2.3 Explanation of object states:

Object states represent, in fact, the values attached to an object. These values describe the current properties of that object. According to the setting of such values, some actions upon the object might or might not be possible to perform. The object states belong entirely to the informational level of the user interface. The possible states are:

I Default: when a user makes no specific changes to the current selections, a default object is the one, that is currently active. This object will be selected for execution, when the user performs an action. The action could be a default selection procedure. This means that the user pushes the enter, or return key on the ASCII keyboard.
II (Non-) Selectable: each object that is displayed, is either selectable or not. If an object is non-selectable it will be displayed greyed-out, which means that it will be displayed in an obviously lighter colour font, than the surrounding selectable objects.
III (Non-) Highlighted: when an object is selected it becomes highlighted, which means that it will be displayed in a more prominent, often more dark colour scale. Optional text labels are then displayed in reversed colour-scaling (e.g. white on black).
IV (Un-) Marked: this indicates if a certain entry, of a couple of check boxes, is active or not.
V On/ off: this state indicates if a radio button is on and thus active. Only one button in a group-box can be set to 'on' at a time.
VI (Non-) Active: an object can be active, which ensures that the actions the user performs can have any effect on this object. When an object is not active, the user actions will, in general, have no effect on this particular object.
VII Busy: indicates that the system is busy with something and therefore cannot react to most users actions at that time. This state will be revealed to the user by a busy cue.
VIII Scrolling or browsing: this state will occur when the scroll-bars are being used. These bars are positioned on the sides and/or below of the list.

12 Listing and description of the basic sound material:

By making use of the guidelines, on psycho-acoustics, cognition and other human factors, a coarse assortment of sounds can be made, that potentionally match well with the intended interpretation or experience the objects or events should elict.For instance:

Sounds that should be continuously present on the back-ground, can be selected on properties, that are especially related to sensational and psycho-acoustical factors.

'Hitting' sounds can be selected according to the metaphorical associations they should elicit. E.g. hitting big objects like windows, can be represented by louder, spectrally more rich sounds than hitting small objects. The small objects can be represented by more dull sounds.

Sounds also can be sorted on basis of the deictic function they should perform, like indicating the busy state of the mouse pointer.

The sounds that are listed serve as basic sound material. Properties like object state can optionally be represented by changes in the parameters of the sound generator. Apart from such parameter changes, like filtering, a sound can also be mutated; it can be time-stretched, transposed or played without attack phase. The different appearences of the sound can then represent different object states or even entirely other objects or events.

In the listing, a distinction is made between one-shot sounds and continuous sounds. One-shot sounds are short sounds that play trough untill their end when they are triggered and not prematurely stopped. Continuous sounds are looped sounds that are sustained untill they are expicitly turned of.

In relation to the different appearances that sounds can have, the continuous sounds are provided with some additional information. This information reflects something of the possible use and the possible different appearances of these sounds. It consists of:

'AT', which indicates wether or not the sound has an obvious distinguishable attack phase. For instance a machine-like sounds can be preceded by a start-up phase, or another sound that indicates the activation of the sound-event. After such an attack-phase, the sound will proceed to a stationary phase. In most cases the attack phase can be by-passed, enabling the sound to start directly in it's stationary phase. If this is possible, it will be indicated by a check-mark under 'ATL', which means alternate attack. By using the sounds with and without the attack-phases, the raw sound material is already multiple applicable.

Furthermore, the sounds are already roughly grouped, according to their general properties and their most obvious representation of the interface objects.

Table 12.1 One shot sounds:

Name:	**Description:**
Sounds, suitable to represent '*appearing / disappearing*' events:	
1 Luxaflex-down 2 Luxaflex-up	Time-compressed and transposed versions of pulling up/ letting down luxaflex.
3 Drawer-open 4 Drawer-close	Time-compressed and transposed versions of the opening/ closing of a wooden kitchen drawer.
5 CD-open 6 CD-close	Transposed versions of the opening and closing of the loader of a CD-player.
7 Airvalve-Escape 8 Airvalve-EscapeReverse	Very high, pretty aggressive sounding release of pressure of a small valve, also in time-reversed version.

9 Elevator-open 10 Elevator-close	Transposed version of the opening and closing of the doors of an elevator.
11 Comy-flute-Down 12 Comy-flute-Up	Slapstick-like nose-flute glissando in falling and rising direction.
13 Comy-Stretch 14 Comy-release	Slapstick-like stretching of elastic material and a reversed version.
15 Elevator-Stop (only for disappear)	Braking sound of a stopping elevator.
16 Servo-Stop (only for disappear)	Time-compressed and transposed version of switching of a running servo-motor.
Sounds suitable for '*select*' events:	
17 Cork-Pop2 (Select-List)	Transposed version of a cork popping of a bottle.
18 Mouse-Click (Select1)	Recorded sound of pushing a mouse button.
19 SashWindow-Down (Select2)	Time-compressed and transposed version of a wooden Sash-Window pulled down.
20 WalkyTalkyCheck (Select1)	Noisy 'check' sound, resulting from switching of a communication channel of a WalkyTalky.
21 AirEscsape	Air escaping from a big opened valve.
22 WindScreen-Wiper (Hit/Select)	Modified version of one cycle of a wind screen wiper, heard from inside a car.
23 FootHat(i[1]) (Hit/Select)	Transposed version of opening and closing a foot hit-hat of a drum kit.
24 ComyHit (Hit/Select)	Sound of hitting a small plastic empty pot, or quickly pulling of it's pot-lid.
25 ComyCrash (Select-List)	Tin-like slapstick sound of a small collision.
26 CD-Insert (Select2)	Putting a compact-disk on the player's loader.
Sounds that are suitable for hitting events:	
27 ComyHit1 (small)	Short, dry hit, like on a cardboard-box.
28 ComyHit2 (big)	Hit, like on a big empty plastic bottle.
29 ComyHit3 (small)	Short, high, metallic hit.
30 ComyHit4 (big)	Hit, like on a washing tub, filled with very little water.(A little 'splash' can be heard at the end).
31 ComyHit5 (small)	Automobile horn, high and short
32 ComyHit6 (big)	Autohorn, little thicker, lower& longer.
Sounds, suitable for '*scraping*' imitations:	
33 Scrape	Scraping of a shovel over the pavement.
34 Scissors (Shears)	Quite low and grinding cut with a pair of shears.
35 ComyScrape	Sounding a bit like a rattle-snake

Table 12.2 Continuous (looped-) Sounds:

Name:	Description:	AT	ATL
1 SoftelPiano(i)	Bell-like piano sound	X	
2 HardelPiano(i)	More raw sounding bell-piano, containing more inharmonics.	X	
3 Watch-Single	Thickened Simulation of a tick of a watch clock.	X	
4 Watch-Double	Similar as above, however every clock-pulse consists now of two cascaded ticks, with emphasise on the first one.	X	
5 Beeb	Beeping of an electronic alarm clock.	X	
6 FM-noise	Noisy signal of an FM radio that is tuned between two channels.		
7 Electricity	Shortly looped sound of a welding apparatus.		
8 Electricity LP	Shortly looped, transposed and low passed filtered, crispy sound of electricity, with sparks jumping over.		
9 Electricity HP	Same sound as described above, however here HighPass filtered		
10 Airbrush	Sound of a small air-sprayer used for e.g. cosmetics or graffiti.		
11 PianoDrone	Modified+looped version of the harmonic sound that is produced by knocking on a piano's sound board.		
12 AM-Partials(i)	Synthesized, bell-like sound in various roughness appearances, depending on the amount of added harmonics introduced by waveshaping techniques.		
13 SpaceRadioStation (i)	Synthesized simulation of a science fiction-like atmosphere.		
14 SpaceRoom	Humming atmosphere of the environment in a control-room of a space vehicle.		
15 SpaceRoom+Blieb	Same atmosphere as described above, but with additional computer bleeps		
16 EnigmaPatch(i)	Modified, randomized and highly resonating sound of a crash-backen, resulting in an SF-like effect.		
17 Comy Ringing	Slapstick sound, like turning a wheel of fortune		
18 Comy Percusloop	A drumming loop on a small, smooth sounding wooden percussion object.		
19 Telephone-Busy (i) (pointer)	Synthesized imitation of the beep-sound that can be heard, when a telephone in use, is called.	X	
20 Magnetron (pointer)	Quick start, followed by stationary sound of a magnetron oven in use. A light bell-'ping' occurs when the sound is stopped.	X	X
21 Ventilator (dial)	A time-compressed and transposed version of the start and stationary sound of a fan.	X	X
22 Servo-On + Stat.	A time-compressed and transposed version of starting up a servo motor, followed by it's stationary phase.	X	X
23 WalkyTalkyCheck	Sound, similar to the one-shot WalkyTalky sound, however here in a sustained (shortly-looped) version.	X	X
24 DentistAirLoop	Airy sound, like that of a high-pressure air-sprayer/sucker used at the dentist for cleaning your mouth, while removing for instance tooth scale.	X	X
25 Elevator+ting+ Stat.	Indication of arrival by a tiny bell-sound, followed by closing the elevator doors and a constant humming sound.	X	X

26 Match-burn	Modified version of lighting a match, followed by a steady burning phase.	X	X
	Some steady sounds, very suitable to serve as background:		
27 DDS-birds	Atmosphere of a half-open railway station, with birds in the back ground.		
28 Fireplace	edited+looped version of a fire in an open fireplace.		
29 Birds	Birds outside heard trough the open window (no traffic in the background).		
30 Wood/Wind/Birds	The atmosphere in a wood, with the wind blowing through the leaves and birds in the back ground.		
31 WoodWind	Only a steady wind blowing through trees&leaves.		
32 Crickets	Steady, quiet outdoors atmosphere, with crickets in the background.		
33 Noise(i)	White noise-bands filtered in various ways		

[1] The indication 'i', between parentheses, that can be found behind some of the sounds, only means that the source for this sound material was the Kurzweils internal ROM sample memory.

13 Objects & Sounds, listing of the test-sets:

If the properties of the interface-objects, events and of the desired sounds are known, sounds and objects that fit well to eachother can be brought together. Lists of sound sets, that represent the interface items, can then be set up. For convenience, the sound set's are subdivided into families of objects or events, that share somewhat similar properties or have similar functions within the windows environment. As a result, 7 families were set up, indicated by the capital letters A to G. For every object and event in these families three sounds are provided.

In relation to the listing of the sound sets, a few terms have to be explained. *LF* means low-pass filtered: the higher contents of the spectrum are filtered out, so that the sound will be more dumb. However, these filtered sounds will not be much softer, because the amplitudes are adjusted in such a way, that the original and the filtered sound will be perceived almost equally loud. [+octave] indicates that, both the original sound and the same sound one octave higher, will be played together. Transposing the second sound up one octave, results in it's frequencies being doubled. (+3ST) means that again the orignal sound together with it's transposed version will be played, but the transposition is only three semitones upward. The amount of transposition needed, depends on the properties of each sound. For some sounds, even small transposition intervals can already produce obvious audible effects.

To save space, the names of the sounds are sometimes abbreviated. In these cases, there is often a fully out-written version in the neighbourhood as a reference. The names cover most part of the contents of the sounds, although some names are chosen rather intuitively, or according to the associations the sounds elicited. Providing the sounds with proper names is quite difficult, especially for sounds that seem to have no obvious connection to sounds from reality, or that are even completely synthetic.

13.1 Listing of the objects, events and sound sets:

KeyMap A:

Object	State	set1	set2	set3
Button	Selectable	/SoftElPiano	PianoDrone	/AM-Partials
	NSelectable	/" *LF*	" *LF*	/" *LF*
	Def/ Selectable	/" + [octave]	" + [octave]	/" + [octave]
Check-Box	Unmarked	Watch-Single	ElectricityLP	DentistAirloop1
	Def-Unmrkt	Watch-Double	" + [octave]	DntstAirlp2(+3ST)
	Marked	WSingle+Bieb	ElectricityHP	DAir1 +WTChck
	Def-Marked	WDouble+Bieb	" + [octave]	DAir2 +WTChck
	NSelectable	Watch-Single *LF*	ElectricityLP *LF*	DntistAirlp1 *LF*
Radio-Button	Off	FM-Noise	Match-Burn	/SpaceRstat
	On	"	"	/"
	Def-On	" + [octave]	" + [octave]	" + [octave]
	NSelectable	" *LF*	" *LF*	" *LF*

KeyMap B:

Object	State	set1	set2	set3
Edit-Control	Active	DDS-Birds	Wood/Wnd/Bird	Crickets
	Non-Active	" *LF*	" *LF*	" *LF*
Edit-Line	Hit/Select	WindScreenWiper	/FootHat	Comyhit
Text-Line	"	"	"	"
List-Box	Above	DentistAirloop1	Wood-Wind	ElevatorStationar
(as PD)	Appear	Luxaflex2Dwn	CD-Open	Elevator Open
(as PD)	Disappear	Luxaflex2Up	CD-Close	Elevator Close
List	Scrolling	?[1]	?	?
	Stop+Select	Cork-Pop2	CD-Insert	Comy

KeyMap C:

Object	State	set1	set2	set3
Menu-List	Appear	Drawer-Open	AirValveEscape	ComyFlute Down
	Disappear	Drawer-Close	AirValEscReverse	ComyFlute Up
	Hor.M.List	?	?	?
Menu-Item	Non-HighLgty	FirePlace	DentistAirloop1	AirBrush
	Def	Match-Burn	DntstAirlp2(+3ST	" + [octave]
	NSelectable	FirePlace *LF*	DntistAirlp1 *LF*	AirBrush *LF*
PD-Edit	Non-HighLgt	Birds *LF*	Ventilator *LF*	SpaceRoom
	Def	Birds	"	" + Bliebs
Icon	Non-HighLgt	PianoDrone *LF*	Airbrush *LF*	Wood-Wind *LF*
	Def	PianoDrone	Airbrush	Wood-Wind

KeyMap D:

Object	State	set1	set2	set3
DialogueBox	Appear	Elvatr.ting+Stat	Servo on+Stat	Comy dial
	Above	(stationarphase)	(stationarphase)	Comy ringing
	Disappear	Elevator Stop	Servo Stop	Comy dial
Window	Appear	/Noise	/Noise	/Noise
(See sepe-	Disappear			
rate Window	Active			
discussion	Non-Active			
for details)				
TitleBar	Above	*Vibrato* (max)	*Vibrato*	*Vibrato*

KeyMap E:

Object	State	set1	set2	set3
Group-Box	Inside	*Tremolo* (max)	*Tremolo*	*Tremolo*
Graphic	Above	Electricity	FirePlace	*i* EnigmaPatch
Pointer	Busy	Magnetron	Telephone Busy	ComyPercLoop
		Ready= KeyOff⌋		

KeyMap F:

Object	State	set1	set2	set3
Hit1	Hit General	Comy1	Comy3	Comy5
Hit2	Hit FamD	Comy2	Comy4	Comy6
Select1	Sel cursor	MouseClick	WTChck	Short-Scissors
Select2	Sel Default	SashWindowDwn	AirEscape	CD-Insert
Dragging		Scrape	Scissors	ComyScrape

KeyMap G:

Object	State	set1	set2	set3
AdGuidance	Distance/Direction	*i* ?	?	?

Table 13.1. Family A up to and including G.

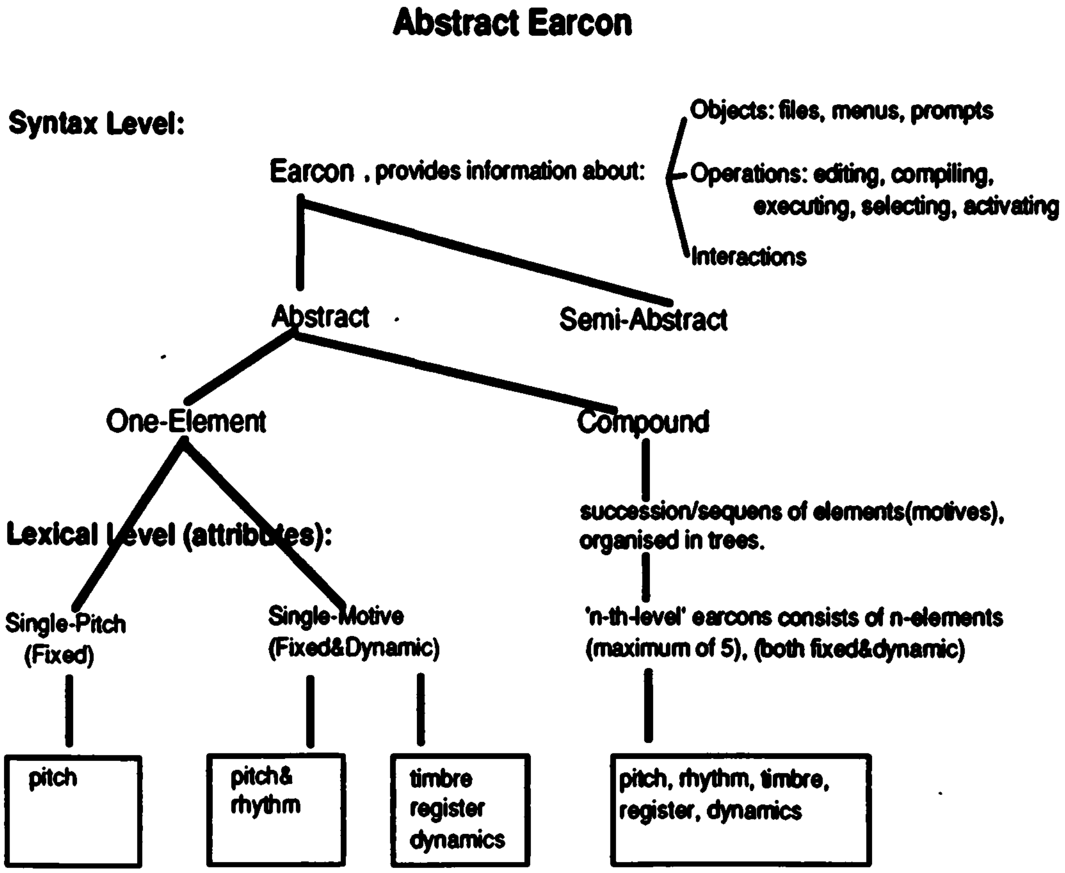
[1] The cells that are indicated with a question mark, are the interface objects or events that are not yet implemented in the current system. More on that topic will be discussed in chapter 16.

14 Earcons, their structure and classification:

In chapters 9&10, only abstract sounds, denoted to as audicons, were handled. As mentioned before, other approaches for the auditory representation are also possible. One of them is by using short pitch sequences, that are called earcons. Because of their somewhat different nature, as compared to the audicons, the earcons sound set will be discussed as a separate issue. In the previous chapters a classification of interface objects and events in families was used. In a similar way, earcons-sequences can also be grouped into families that share aspects in common.

Recognisable family properties for earcons are:
1 Timbre,
2 Absolute Pitch; with 'absolute' is meant the exact pitch of a certain note, whereas
3 Pitch-Region means; the actual pitch range that is covered by the earcon-sequence,
4 Pitch-Melody; the actual note-sequence that is assigned to an earcon,
5 Rhythm; the rhythmical attributes of the sequences,
6 Total-Number-Of-Notes; this is a very recognisable aspect, e.g. if a sequence consists of 3 or 4 notes, is an easily distinguishable fact,
7 Interval-Structure; for the intervals that are used, the earcon guidelines have to be taken into account. Other aspects like streaming effects, depending on the rate of presentation (tempo) and the interval structure, also apply here,
8 Dynamics; also the dynamics have to be applied according to the earcon guidelines.

The definition of earcons and their appearances have already been mentioned in chapter 2, §2.2. However, to gain some insight in the way that compound earcons can be created, a systematic overview might be clarifying (fig. 14.1):



Abstract Earcon
Syntax Level:
Earcon , provides information about:
Objects: files, menus, prompts
Operations: editing, compiling, executing, selecting, activating
Interactions
Abstract
Semi-Abstract
One-Element
Compound
Lexical Level (attributes):
succession/sequens of elements(motives), organised in trees.
Single-Pitch (Fixed)
Single-Motive (Fixed&Dynamic)
'n-th-level' earcons consists of n-elements (maximum of 5), (both fixed&dynamic)
pitch
pitch& rhythm
timbre register dynamics
pitch, rhythm, timbre, register, dynamics

Fig. 14.1

Earcons can be build up in levels. Each level represents one of the recognisable family properties. The **first level** consists of the characteristic rhythms:

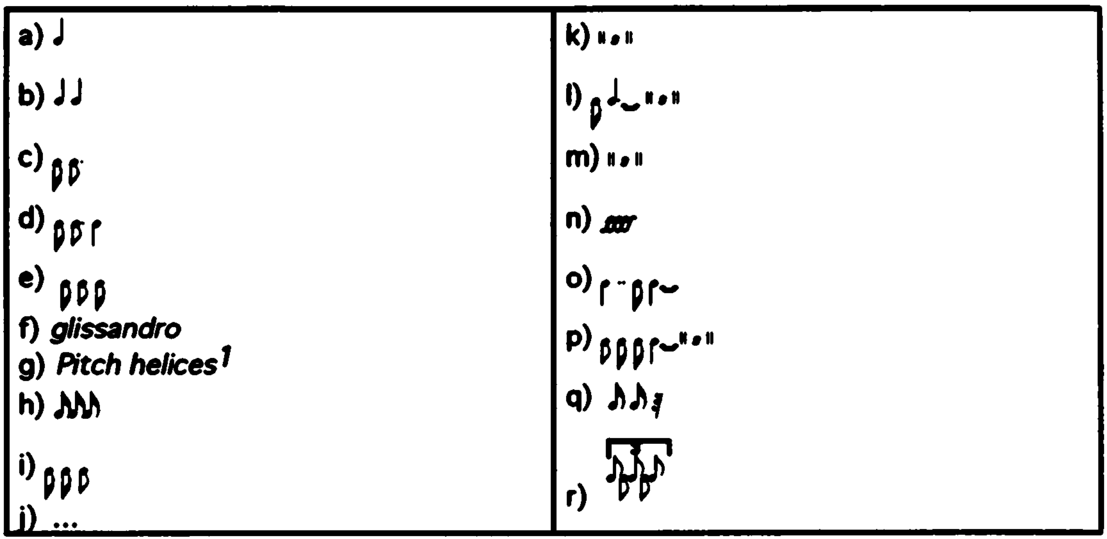


Table 14.1

[1] Pitch helices are sound paradoxes. While the pitch of a sound seems to rise or fall stepwise or continuously, it is in fact never getting really higher or lower. Such an effect can be compared to the visual effect of a never ending waterfalls, as painted by Rudolf Escher. In these drawings, the water seems to flow down, but in the end it is still at the same level as where it started from.

At the following, **second level**, pitches can be attached to the rhythms. With the **third level**, the used timbres are defined. This level provides the most obvious indication of the family to which each object belongs. Thus, the objects are classified into the most easily distinguishable earcon parameter, which is timbre. The interface objects and events are subdivided into families in exactly the same way as with the audicons. Here, the classification by timbre of the earcons, is done in instrumental groups. This family classification in instrumental groups is listed in table 14.2:

FamA: Bell-like sounds	FamB: Strings	FamC: Organ	FamD: Synth-Layers
FamE: Brass	FamF: Percussion		
The Auditory Guidance and tremolo effects for e.g. group boxes, are used in the same way as with the audicons.			

Table 14.2

Optionally included dynamics are introduced at level four. Dynamics can be used to emphasise certain melodic or rhythmical structures. At the final, **fifth level**, modifications to the appearance of the earcon as a whole are described. In the current project these modifications are limited to filtering of the earcon sounds. In this way there is provided for compatibility with the audicons, where a similar technique is used. Other possible modifications are expanding or compressing the earcon sequences in time, or transposing the entire sequence in pitch. The restriction is that the modifications should operate on the earcon-sequens as a whole and that the sequence remains recognisable, as compared to the original.

Tempo:
Apart from the earcon guidelines, described in chapter 7, some other aspects have to be considered when designing earcon sets. Tempo is one of these aspects. Sequence rhythms should be related to the tempo, that will be used for playing back the earcons.
The tempo could be defined by referring to the guidelines for the smallest note-lengths that should be used. The suggestion is not to use note lengths smaller than 0.125 [sec]. The biggest note length that can be used is not so precisely determined. Anyway, when the total sequence length, it's effect on the speed of operation and human factors like auditory memory are considered, it should be possible to determine a limit for the maximum note length as well. In some studies on the cognitive psychology of human audition, (Crowder) found that recognition of recycled (looped) melodies are possible at a range of rates, that cover an order of magnitude from ± 200 [ms] per tone to 2 [s] per tone. Best results were obtained with rates of ± 320 [ms] average per tone.
This corresponds to a pointed quaver at ±120 [Bpm]. At that tempo the smallest note value that should be used is a 16th. When keeping in mind the best average note-value of ♩. at 120 [Bpm], a useful biggest note-value can be a halve note.

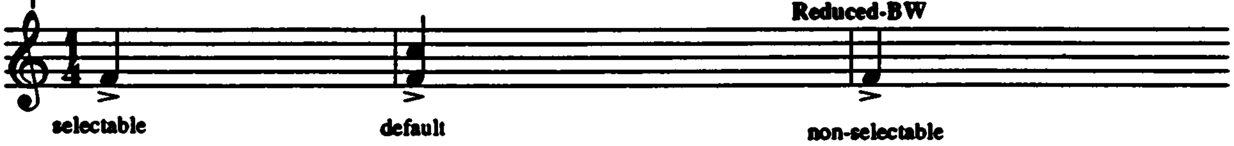
Whole notes and longer can then be reserved for explicitly sustained notes. The function of these sustained notes can be compared to the continuous sounds, that are used with the audicons. By using the level classification of the earcon families and the additional guidelines, some earcons sequences where created. These sequences are listed in the next chapter; chapter 15.

15 Earcons, listing of the earcon motives and their additional parameters:

In general the event-sequences are triggered when an object is encountered by the mouse-pointer, although in some cases the events occur as a result of a user action or system state change. After a sequence is played through, the last note will be sustained as long as the pointer is still above the object. The sustain section of the sounds should be of low amplitude to avoid distraction and listening-fatigue. To accomplish this, every used earcon sound has an amplitude envelope, that is especially adapted for that earcon.

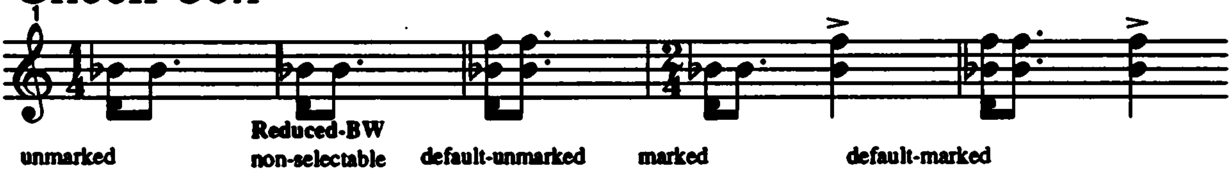
FamA:

Button



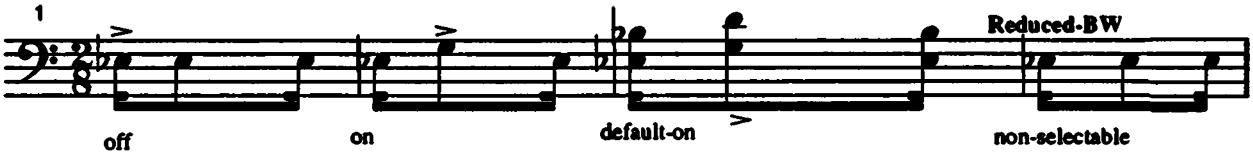
Sound: Grandpiano

Check-box



Sound: SoftElectricpiano

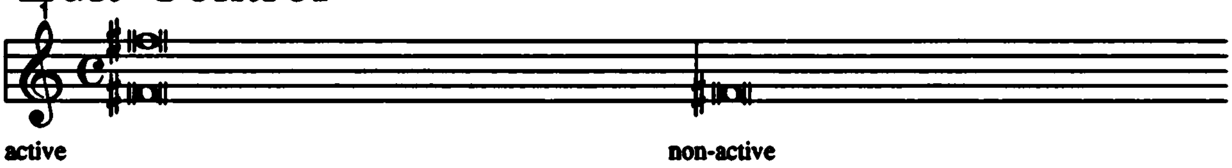
Radio-Button



Sound: HardElectricPiano

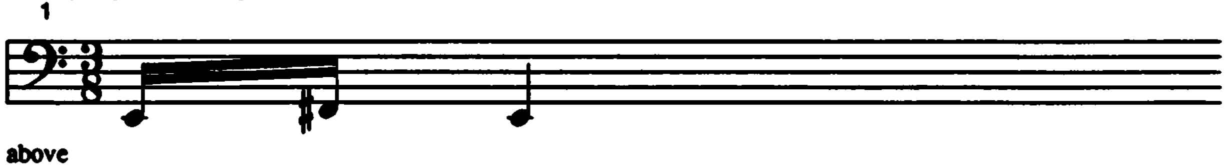
FamB:

Edit-Control



Sound: GrandStrings

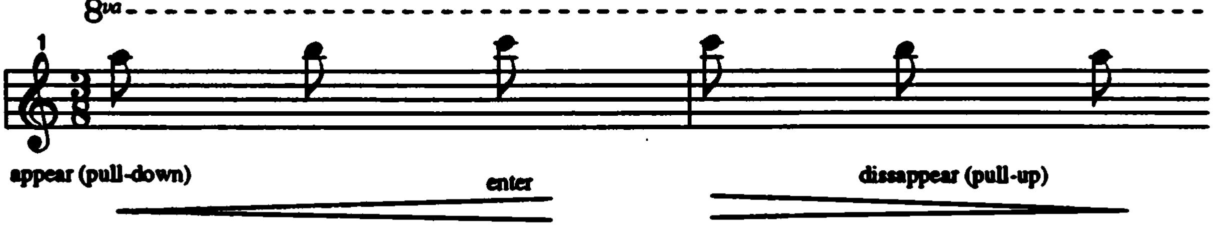
Edit-Line



Sound: EnsemblePizzicato

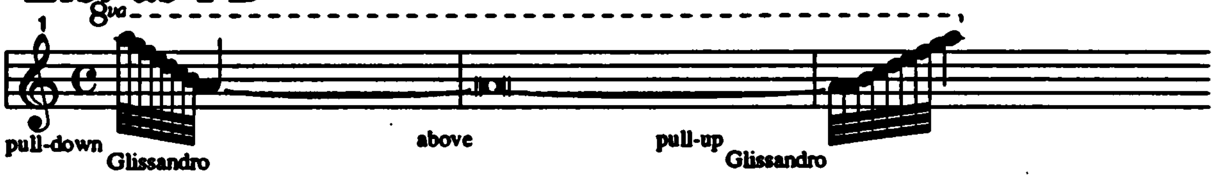
Text-Line has the same sequence as the Edit-Line. Although there is a difference in function between these two kind of line, the visual appearance is similar. For this reason, the auditory representation for both line-types, will be the same as well.

List-Box



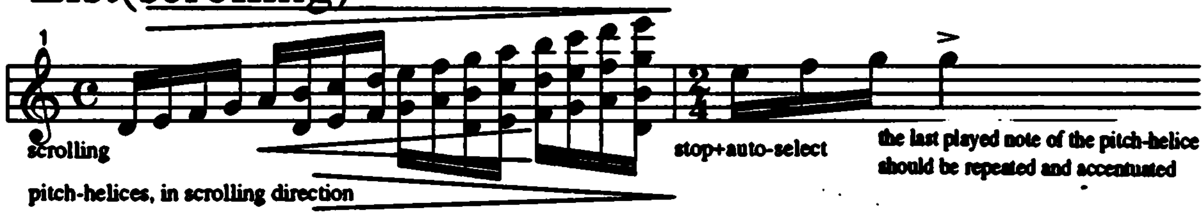
Sound: TouchOrchestra

List-as-PD



Sound: BarokStrings

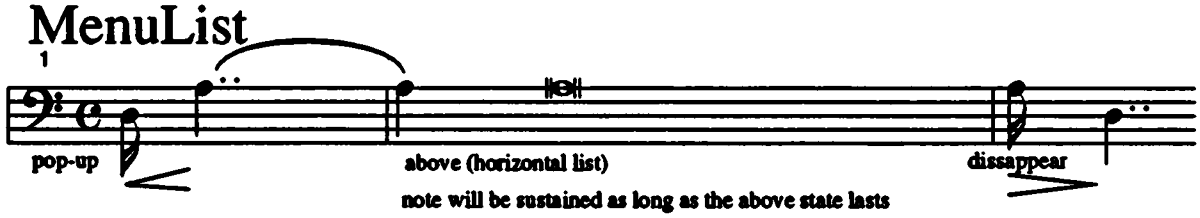
List(scrolling)


1
scrolling
pitch-helices, in scrolling direction
stop+auto-select
the last played note of the pitch-helice should be repeated and accentuated

Sound: Strings

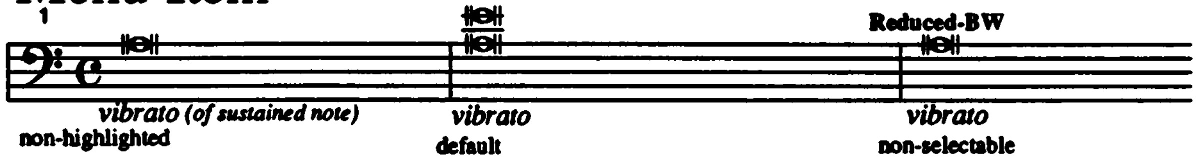
<u>FamC:</u>

MenuList



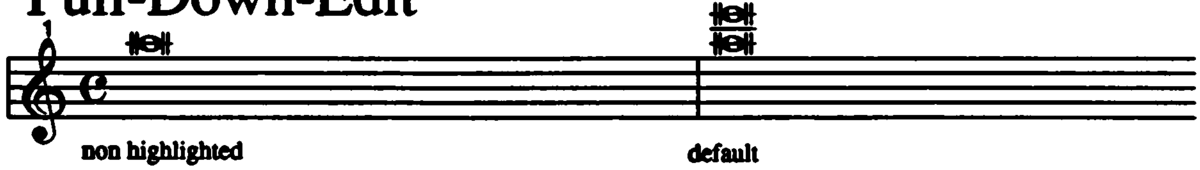
Sound: GospelOrgan

Menu-Item



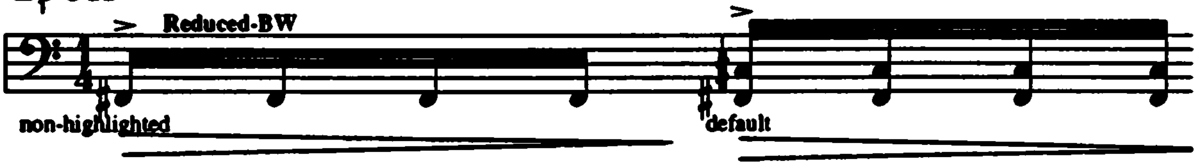
Sound: GospelOrgan

Pull-Down-Edit



Sound: BalladOrgan

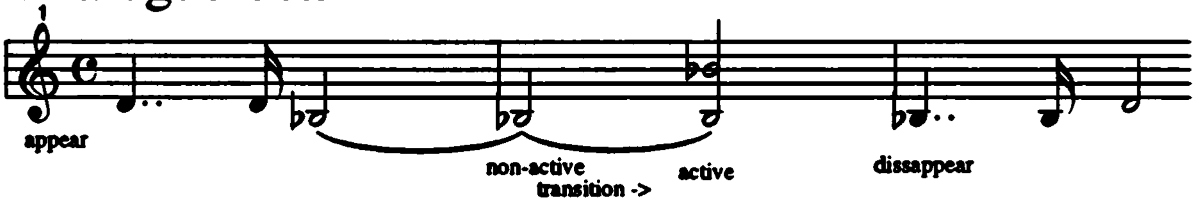
Içon



Sound: PercussionOrgan

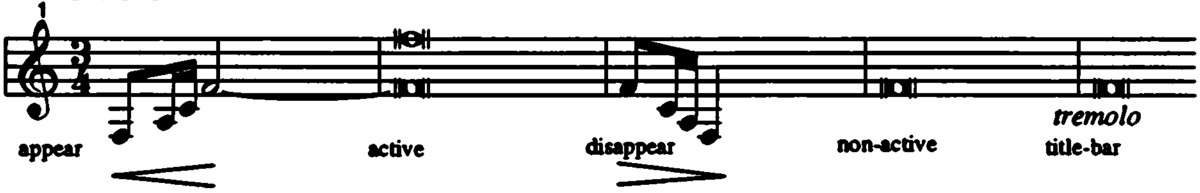
FamD:

Dialogue-box



Sound: Fantasia (SynthLayerSound)

Window

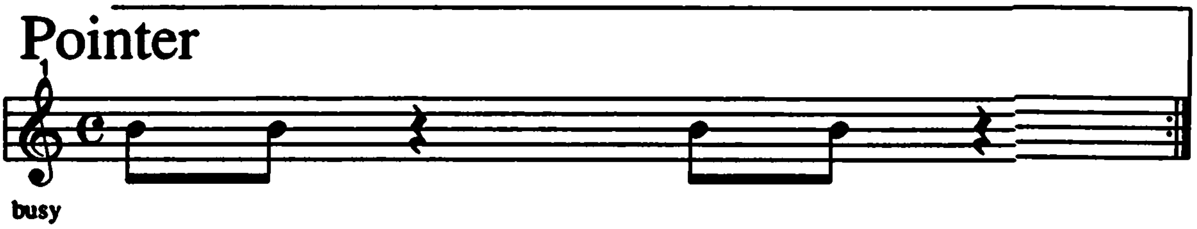


Sound: Tranquillity (SynthLayerSound)

FamE:

If the pointer is positioned within the outlining of a GroupBox, will be presented by a tremolo effect in the currently sustaining sound. If the pointer is above a Graphic is indicated by a noisy-background sound, that has no specific earcon sequence assigned to it. In a sense, this sound could be seen as an audicon, but here it is interpreted as a one note earcon sequence. The rest of the 'E'-family consists of:

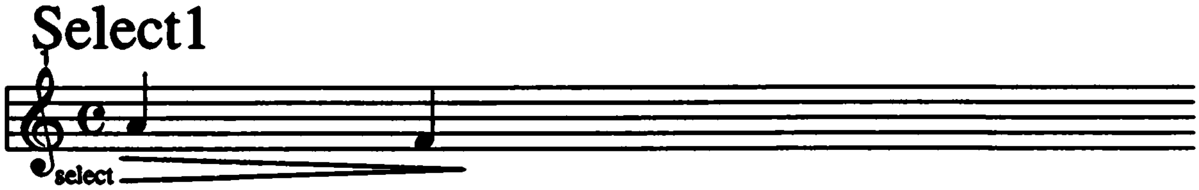
Pointer



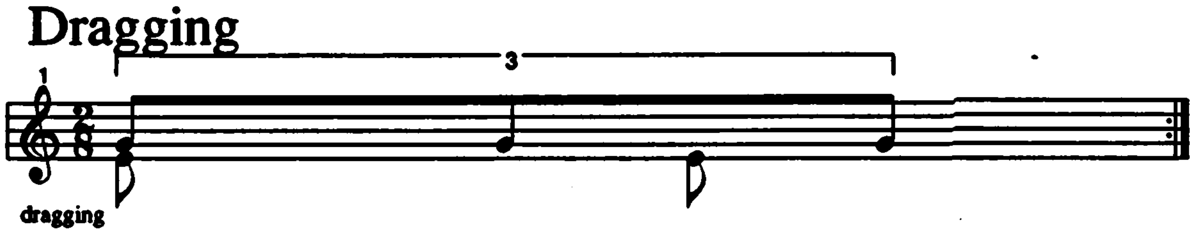
Sound: TenorSax

EamE:

As far as the event sounds are concerned, percussive sounds are most representative for these events. For instance, the Hit sounds are represented by one note sequences of the following percussion sounds Hit1: Clang hit and Hit2: timbale hit. And further:



Sound: Pitched PercussionSound



Sound: Cabasa (PercussionSound)

The Auditory Guidance is a universal aid in the navigation. It is therefore not bound to the kind of auditory representation that is used (audicons, or earcons).
The specifications of the AD-Guidance will thus be the same as those that are used in the audicon approach and can be found in §17.2.4.

16 Restrictions related to the practical implementation of the sound cues:

As always during a project, some problems can occur, or limitations on the implementation of designs can be encountered. Some of the restrictions that had to be taken into account will be discussed in this chapter. The limitations were often related to properties of the hardware that was used. Such restrictions led in certain cases to compromises. These compromises will be listed in this chapter.

16.1 Limitations related to the screen recognition system:

Some objects or events could not (yet) be recognised and/ or translated by the system that should process the current screen contents for further use. Due to this fact the following objects or events could not be used or tested by this time:

Scrolling: the scroll bars and scrolling actions are recognised, but not yet implemented in the current system.

Group Box: The outlining, indicating that a set of encircled objects belongs together, is denoted to as a group box. The screen recognition system is not yet able to recognise these group-boxes. Therefore group_boxes are not included in the test sets.

16.2 Limitations related to the sound playback hardware:

For the play-back of the audicons and earcons a MIDI-synthesizer/ sampler was used. This put certain limitations on the sound synthesis possibilities. For instance the resolution of many sound parameters was restricted, due to the limited resolution of MIDI. Especially when psycho-acoustical factors were considered, these limitations led to certain compromises. For instance loudness-curve-compensation was only partly possible by using a coarse estimate of the real curves. Such curves are not standard to the used synthesizer and for programming simulations, almost only linear curves and parameter ranges, consisting of 127 steps, were available.

Another aspect of the use of MIDI is that every intended sound parameter setting or modification has to be translated and estimated from real world parameters to MIDI-parameters and v.v. Real world parameters consist of frequency- and bandwidth-indications in Herz and amplitude proportions in Decibels. MIDI parameters, on the contrary, are assigned in key-numbers, velocity values, etc. The formulas and estimates that were used for the necessarily translations can be found in appendix B.

A third limitation of the use of a MIDI synthesizer/ sampler was that the possible sound modifications were limited to the functions that are build-in this MIDI-equipment. Although the Kurzweil K2000, that was used in the experiment, has fairly free configurable DSP functions, these functions have to be chosen from a fixed set. Also, the amount of DSP functions that can be used simultaneously is limited.

Hardware constraints, related to the use of a MIDI sampler/synthesizer are directly related to the most prominent limitation that was encountered in the sound-design process. Analysis and destructive editing of the sound material, according to step 7 and 8 (§ 7.9), was not possible with the Kurzweil. The necessarily steps could have been performed on a computer with the appropriate software. However the software version of the Kurzweil did not allow for transference of the sample data to- and from an external device. As a result, step 7&8 of the design process could only partly be performed. The tasks that could not be performed are:

a) Sounds could not be destructively changed in their spectral contents. The sounds were entirely edited on the Kurzweil, which has no destructive filtering or equalizing functions.
b) The sounds could not be analysed on their exact spectral contents and possible masking effects.
A last option to accomplish this could have been to record the output of the Kurzweil, store the data in a computer and analyse it. Still then, destructive editing according to the results of the analysis would not be possible. Also this option would have been very time consuming and problems with amplitude mismatches during the transferences, would most probably occur. Due to these factors and because of time-limits, this alternative option has not been chosen.

To reduce the effect of the omission of the mentioned tasks, the final balancing of the amplitudes of the different sounds have, as precisely as possible, been done from hearing.

17 Discussion & Recommendations:

Once the sound sets have been created and the drawbacks are known, some cautious predictions can be made on what can be expected from their use in practice.
If it has to be examined how effective the sounds will be in practice, experiments for this purpose, should be conducted. Expectations on the results of such experiments are asserted in the first part of this chapter.

The second paragraph of this chapter contains suggestions for future improvements, or extensions of the auditory interface. Some of these suggestions are only coarse ideas, while others are completely worked out plans, that are ready for implementation.

17.1 Expectations:

The expectations about the practical use of the sounds sets, accesses several aspects of the use of sound. These aspects can be divided the following issues:

Noise: Noisy sounds, like the noise bands used for the windows, could have making effects on other sounds, although they are presented at a rather low amplitude level. This masking is a psycho acoustical factor. Unfortunately, the exact influences of such psycho acoustical processes could not be analysed on beforehand.
This restriction is discussed in chapter 16, §16.2.

Learning: A second fact that has not been discussed this far, is the influence that learning processes can have. In the first experiments there will be no time reserved for a learning period. In this way, the subjects will not have any primary knowledge of the sounds that will be presented. However future experiments might show that learning periods can improve the performance. For sounds that have decreasingly less real-world relationship, thus become more abstract, the importance of a learning period most probably increases. This is especially true for earcons which are in fact entirely abstract.

Listening volume: People might be in need of turning up the listening volume, as long as they are not familiar with the sound set. During this first period, they still have to pay much attention to be able to recognise the sound and the meaning they have within the interface environment. By turning the volume up, they can listen more carefully to the presented sounds. After a learning period, when people become more and more familiar with the sounds, the presentation level should be chosen as low as possible, to enable an almost unconscious processing. In the ideal situation the sound cues are automatically auditioned and interpreted by the user. Compare this to the sound of a car driving by somewhere. The sound and it's meaning are immediately recognised and interpreted, without explicitly having to focus attention upon it and without the need to be presented at a high loudness level. This aspect is thus connected to learning and memory aspects.

Memory: Some experiments will be conducted with the sound sets that are described in this paper. The goal is to extract the sounds that show the best correlation with the interface objects they represent and create a new set from these sounds. In this first experiment, the visual objects and their related sounds are presented in a relatively high speed rate. People will most likely have problems remembering all the events and the relations between the objects and the sounds that represent them. Learning time could again improve the performance, but it overshoots the goals of this experiment. Memory-aspects are not a point of investigation for the first experiments. Still it may prove useful to pay attention to the subject's reactions in relation to such aspects.
In this way, some useful information could be gathered, that can be of use for future experiments, where this memory aspect is included.

Speech versus non-speech output: Another aspect, only sideways mentioned in chapter7, is that next to the non-speech audio cues, also speech output is provided. The speech will be presented at a much higher volume level than the non-speech audio. A decrease in comprehensibility of the speech, due to masking by the non-speech audio is not to be expected. On the other hand, as long as there is speech output, it might blur the non-speech audio. Fortunately, this will not cause a decrease in the understanding of the meaning of the sound output, because at these moments, the speech- and non-speech audio represent the same message to the user. Anyway, it still has to be examined what the best method for the representation of both the audio signals will be. The possible choices are: first the speech-output and then the sound-cue, or vice verse, or presenting both the signals simultaneously.

Orthogonality (independence): While examining, when and how sound events should be presented, another problem arises. Within the interface the events are mostly not completely independent. Often they are a logical result of an other, previous event. .
For instance the action of selecting a button can be represented by a select sound, but simultaneously, the state of the button itself can change, due to the selecting action. This means that next to the select sound, also the sound cue indicating the state change of the button, should be played. Apart from these two sounds, the result of the select action, like the appearance of a window, should be made audible too. Because all these events happen within a very short time-period, choices have to be made. It has to be decided which sound will be played. As an alternative, the delay between what visually happens and the related auditory representation has to be intentionally enlarged, to enable each sound to be entirely played in correct order. In fact the designer has to deal with non-orthogonality in both the possible events in an interface and in the parameters of the sounds.

The use of samples: A property of the use of sampling based techniques for the sound programs, is the exact repeatability of the sounds. When sound are played very often, the exact repetition might start to irritate the user. This may also happen, when the iterations of loops in sustained sounds are very noticeable. In relation to these facts it might in some cases be better to use entirely synthesised sounds, instead of looped samples.

Variations in the sound parameters: In the contrary to the properties of some sampled sounds, people also might dislike sounds that contain to much variation. Sounds that have quickly changing frequency and amplitude contents, especially with large and sudden jumps in it's parameters, could cause increased distraction and even much irritation. Unless these effects are the intended by the designer, such sound programs should be avoided to keep a smoothly balanced auditory interface.

Using cartoon sounds: In the sound sets that have been developed for this project some cartoon sounds were included. Cartoon sounds represent certain actions or events, taking place in a cartoon, by exaggerating some properties of naturally occurring sounds. Cartoon sounds are in fact some kind of caricatures of the natural sounds.
The intention of the sounds is in most cases very obvious. It is therefore very useful to examine the techniques used by sound-designers working in the cartoon industry. Perhaps some of them can be applied to the sound programs for the auditory interface. Still the experiments have to show if cartoon-like sounds will meet the expectations.
It could be that they will not be excepted by the users, because the sounds are not taken seriously, or are considered to be to childish.

Sound cues used in radio and TV broadcasting: The use of cartoon sounds, is slightly connected to the use of sound cues in radio and Television broadcasting. Especially in quizzes, many sound cues are used to indicate; good or wrong answers, the development in tasks that people have to perform, the elapsed and remaining time, etc.
As with the cartoon sounds, much experience on how to transfer information to people or elicit certain affects with sound is collected. Although this field was not examined for the present sound sets, it might prove very useful to do so for following projects.

To conclude the first part of this chapter, I hope that more experiments will be conducted in the future on:

The usefulness of each specific sounds, in relation to the object or event that it represents.

The exact physical requirements that sounds should meet, when they are to be used in an auditory interface.

The performance of auditory interfaces in real practice situations, when the complete interface-environment and some applications are implemented.

Such research projects will probably deliver more usable results, than projects that only address the general question, if sounds can be used to represent an interface environment. The question of exactly which sounds are to be used for which event seems much more important to me.

17.2 Suggestions for further integration of sound in the interface structure:

In this second half of chapter 17, I will describe some ideas that have not yet been implemented in the current auditory interface, but which can be eventually used in future extensions to the system. Some of the suggestion only consist of , more or less, personal ideas, while others are completely worked out plans. Optionally these plans can be implemented and tested on their usefulness.

17.2.1 Suggestions related to some human factors :

Memory:

Especially in the beginning, people will probably have problems with memorising all the exact meanings of the sound cues. An option could be implemented to help the user in this purpose. A mouse-button or key on the ASCII-keyboard could be reserved, to enable the user to retrieve an explanation of the meaning of the latest heard sound, by pushing this button or key.

Irritation:

To avoid irritation, due to exact repetitions of sounds, slightly random variations in the sound parameters might give a more natural quality to some sounds. Users might experience this as being more pleasant to listen to.

Comfort:

For listening comfort, the sound playback quality should preferably be CD-quality. . Especially when spatialisation techniques are considered, for future implementations, a good sound quality will be necessarily to obtain proper results. It also might show that users will get less quickly tired of the sound, when the audio-quality is increased.

17.2.2 Suggestions related to the inclusion of spatial information:

Foreground/ background:

Objects that seem to have a position, more or less, in the background of the visual interface, could also be placed further away in the auditory representation. For this purpose reverberation techniques could be used. For instance windows that are obviously behind other windows, or objects on the desktop that are behind an open (application-) window, could also be spatially positioned further behind, by varying the direct-/ diffuse sound proportion and adjusting the amplitudes.

'Empty objects':

Another use of reverberation could be the indication that certain files or objects are more or less empty. The amount and the quality of the reverberation, that is added to the object's sound, could be varied for this purpose.

Spatial arrangement:

In relation to the use of reverberation techniques one should realise that reverberation can provide the blind user with information about aspects like room size, but the actual arrangement of objects within an enclosed space might be more difficult for blind people to imagine.

Headtracking:

If spatialisation techniques are applied, like binauralisation, improved results can be obtained, if a headtracker device is used. Also volume and filter compensations, according to the intended distance of objects should not be omitted.

17.2.3 Suggestions for the representation of windows :

Sound-shadowing:

A physical property of a sound wave is that it cannot bend around obstacles, whose size is longer than the wavelength of the sound. The obstruction of the sound for certain frequency bands is called 'sound shadowing'.

This effect could be used to provide the user with information about the spatial arrangement of objects within a GUI. If certain objects are partly covered by other objects, this could be made audible by introducing filtering effects in the sound of the partly covered object. Such filtering effects can be based upon the sound shadowing theory. The upper object will be obstructing the sound waves, that are 'emitted' by the underlying object. Best results with this technique can be obtained with noisy, or broad-band sounds. With such sounds, the introduced changes will be most obvious.

For instance, the continuous sounds, among the audicons, are often chosen with a broad-band or noisy spectrum. Because of this, they will move into the background of our consciousness, when they are sustained over a longer period of time. As a result, they will less likely start to distract or irritate the user over a longer period of time.

To implement the sound-shadowing technique, it is necessarily to establish a relationship between the physical properties of the screen contents and the characteristics of the auditory domain. To do this, a sensible frequency range should be chosen, that covers the entire screen range. Then, formulas, that are necessarily for the translation between indications in pixels and in frequencies (Herz) can be set up. According to the amount of overlap between objects, it will further be possible to calculate the required filter curves.

Usually when windows are stacked above eachother, the most upper window is the active window. However, several windows can be simultaneously distributed over the screen, without overlapping eachother. Still, only one window is the active window, but all windows appear to have the same stack level. To avoid confusion, it might be useful to introduce an extra sound cue in such situations. This cue should indicate explicitly, which window is the currently active one.

When a sound modification technique is used, to indicate the stack level of windows or other objects, Just Noticeable Difference properties of our hearing system should always be taken into account. Changes in the sound should at least exceed these difference limens, to make sure that they can't be missed by the user.

17.2.4 Suggestions for the use of navigation cues :

For aiding navigation and orientation, within the auditory interface several approaches are possible. Previously some experiments have been done with a pulsating signal, that indicated where the position of the most close object was. This 'Auditory Guidance', as it was called, was based on the paradigm of the blind stick, that blind people use to find their way and to locate obstacles. When an object is hit with this stick, the repetition rate of the 'ticks' indicates the distance of the object. A similar approach was used for the Auditory Guidance. When the object distance decreased, the repetition became faster and to indicate the vertical distance, changes in pitch were used. To indicate if the object was left or right of the pointing device, the signal was presented either entirely at the right ear, or at the left one. With regard to this approach, I have some suggestions that at worth to be considered and that could lead to some improvements of the current system. With some of the ideas, comments, or advantages and disadvantages of the suggestions are also given.

I Using a fixed playback-rate for the pulses of the Auditory Guidance :

Instead of speeding up and slowing down the repetition rate of the sound cues, the lengths of the sounds themselves could be adjusted to indicate the distance-changes. The presentation rate can then be kept constant. The sounds can be varied in duration, from ± 10 [ms], until they form one continuous sound. (No gap in between the sounds).

Pro's: Less distraction and irritation, which is caused by the continuos variations in the playback rate of the sound cues. Such variations can also work very tiring on the long run.

Con's: The changes in duration are more difficult to perceive than changes in tempo and small changes might not be noticed at all. Also when often repeated sounds, become longer than ± 200 [ms], the effect of 'listening fatigue' starts to increase (Rösing).

II Using a fixed playback-rate, but alternating pitches and panning positions, that indicate the absolute positions of the pointer and the objects.

As an alternative, tones with alternating pitches can be used. The first tone can have a fixed, reference pitch, while the pitch of the other tone depends on the objects relative position. The direction of the produced interval, (up or down), indicates the vertical direction in which the object should be searched. The size of the interval will give an indication of the distance at which the object can be found. If the object is at the left or right is presented by panning the sound, according to the averaged horizontal position of the object. Optionally the amplitude can be adjusted, according to the distance of the object. In this way the signal-level of the object will decrease, when the pointing device is moved away from it and vice versa. Unfortunately, dynamically changing the amplitudes can lead to problems that are discussed in appendixA, §3.4.
It is therefore not recommended to use such loudness variations.

To establish a convenient relation between the actual sizes and distances, displayed on the screen, and the presented pitches, some basic rules can be put up.
The maximum, total amount of objects that can be vertically displayed, can be calculated. One could look at the minimal vertical size, expressed in pixels, that objects usually have. Most often, the part of objects that is vertically visual, is at least 8 [pixels] in size. According to this fact, a step resolution of 8 [pixels] can be chosen. This will result in a grid, that is vertically divided in areas of 8 [pixels] in size.
Each grid line will have it's own pitch attached to it. The lowest pitch will be at the bottom of the screen and the pitch increases for every grid line that one goes up.

The total amount of objects that can be vertically displayed on a VGA screen is 60. In the auditory domain, this will correspond to an arrangement in pitches that covers 5 octaves. A sensible pitch range for these octaves will be mid-range.
Summarised: the pitch-grid, covering the vertical arrangement of the screen contents, consists of absolute values, that each describe an absolute vertical position. Each first pitch of the navigation signal represents the current position of the pointing device, while the second pitch indicates the vertical position of the main point of the closest object.

This approach is slightly different, than was announced at the beginning of this paragraph. The pointing device is here no longer the reference tone. The tone representing the object has in fact more of such a function. The object remains at it's absolute position, while the pointing device moves around to find it. In this way, the pitch and panning of the pointing device will change, until they match those of the object. Only the amplitude of the pointing device's tone will be constant, while the amplitude of the object's tone may vary, according to the distance the pointing device has to this object. In outline: The pointer-device-tone has a varying pitch, varying panning position and fixed (reference-) loudness level. The object-tone has a fixed pitch and panning position and a varying loudness level.

III Using the Guidance on demand and avoid the confusion in the definition of the nearest objects.

A third approach for aiding navigation, is using a combination of mouse manipulations and cursor keys. When the user is confronted with a screen full of objects and windows, the cursor keys can be used to jump from object to object.
The starting point will be the current position of the mouse pointer. The user can push the up-, down- , left- , or right cursor key. The system will then indicate the first object appearing, respectively; above, beneath, at the left or at the right of the current position of the mouse pointer. If the cursor cannot be moved further in a certain direction, because the end of the screen is reached, or because there is no object left in that direction, a non-speech-audio cue will indicate this fact.

When the user has detected the object that he is looking for, the Auditory Guidance can be activated. In previous implementations, the Guidance would always point to the nearest object. Now, the Guidance will only point at the object, that was currently located by the use of the key-cursors. In this way the Guidance can guide the user straightway to the object, that he/ she wants to go to.

Summarised: the cursor keys are used to locate the specific object, where the Auditory Guidance should point at. This object will not be really selected or highlighted (as would be the case if the mouse pointer itself was moved onto the object).

The advantage of this approach is that the Guidance will always point straight towards one specific object. It will not change in its indication, every time it passes another object, when this object is closer to the mouse pointer than the one that was currently the target.

Other actions, like dragging objects to a trash-can, can also be simplified in the same way. When one is positioned above an object that should be thrown away, the screen can be scanned for the trashcan, by using the cursor keys. If this is located, one can select+hold the object at the current mouse pointer position and activate the guidance. The guidance will then guide the dragging action of the user, to the trashcan.

IV Some additional notes, relating to the topic on auditory navigation:

<u>Timbre:</u> Especially when a target object is nearby, one cannot always distinguish what the 'first'(pointer-) and the 'second' (object-)tone is, of the Auditory Guidance.
The pointer device and the target-object should therefore have each their own timbre.
Perhaps the characteristic sound of the object, can be used for the Auditory Guidance.
In this way the user does not only know where the target object can be found, but also what kind of object this is.

<u>Inertia:</u> It might be useful to introduce inertia in determining what the closest object is. In this way, the Auditory Guidance-cues will show less jumpy behaviour in its definition of the closest object. This is preferable, when the pointing device is routed through a space that contains many objects that are close together.

<u>Timing:</u> An adequate time-interval between the pulses of the Auditory Guidance, is ±370 [ms]. This corresponds to the natural time interval that will result, when people tap groups of four ticks (Fraisse). People might find this interval length comfortable. It best matches the time-interval, that they would automatically choose for themselves.

<u>Movements:</u> With a fixed pulse-, or repetition rate of the navigation cues, the average speed at which the users moves through the interface, should be taken into consideration. If the user is making fast moves with the pointing device, maybe 'skip-frames' should be introduced. This could be compared to swallowing certain syllables in speech-synthesis, when the playback-speed is increased. In fact, skipping of position frames will already automatically happen, when the guidance takes the most current mouse position as reference for it's sound output-parameters. The movements through the screen contents can then outrun the update-rate of the navigation cues.

Sound timbre: The timbres, used for the Auditory Guidance sounds, should be chosen on base of the function they have within the interface environment. It should be examined what amount of urgency they should express and how much of the attention of the user they should demand. Although the Guidance sound should be distinguishable and easy to follow, it should not attract to much attention in that it becomes distracting or even irritating. Sound characteristics, according to the guidelines for a moderate urgency degree (§5.4), should be appropriate.

Absolute reference: Optionally a reference audio-cue, that indicates the exact middle of the screen could be provided in many applications. The pitch of this tone should then be set according to the vertical middle position of the screen. It's stereo position should be set to the center. The loudness level of this tone can be set equal to that of the tone, that represents the pointing device.

Dragging: The techniques of adjusting stereo-position, (loudness-level) and pitch, can in a similar way, be applied to dragging actions. Pitch will then represent the current vertical position of the object that is being dragged. The stereo position indicates it's horizontal position and loudness level it's distance to the middle of the screen.
The middle of the screen can serve as absolute reference as described before.

17.2.5 Discussion of objects and events that are not yet implemented :

Horizontal_Menu_List: It is not yet decided, if there will be a separate auditory representation that indicates if the mouse pointer is above a Horizontal_Menu_List, instead of a vertical one. This extra distinction might prove useful, while a Horizontal_Menu_List often has a large blank region at the right side of the last Menu_Item. The user should somehow be made aware of this fact, when the mouse-pointer is positioned above this empty field.

Minimize, Maximize and Resize: solutions to if and how these window related events can be presented to the blind user in a sensible way, are still under exploration.
Most probably substitutes will be developed, that present these possible user actions in a more appropriate form for the non-sighted users.

Activate: This user action is not recognised as a separate event. It will be interpreted as a double-select action, followed by whatever events are triggered by this activation action. Nevertheless, experimental results could show that users prefer a separate auditory representation for the activate action, next to the select representation.

17.2.6 Some additional ideas for possibly useful sounds :

Sounds related to movements:

Footsteps could be used to indicate movements. The actual speed of movements through the interface, can control the length of the time interval between the foot steps, that represent the movements.

'squeezing' of (aeroplane-) tires could be used to indicate, that the mouse-pointer is quickly passing over an object. This sound cue can be activated, when the user is moving very fast with the mouse pointer over the screen contents.

Simple tone sweep-intervals up or down, could be used to indicate the appearance or disappearance of system prompts.

The sound of *a car, driving over a speed-bump* , or the sound of a *heartbeat*, could be adequate for indicating that a window or another object is being entered.

Sounds for the *trashcan*:
Additionally some sounds for the trashcan are listed. Although this is actually a more application specific object, it is a common item, that can be found in many GUIs[1].
Some sounds that may be useful to represent several different actions and states that are related to the trashcan:

Number of objects:	**Contents:**	**Throwing away:**	**Emptying can:**
single object	document	crumpling up of paper	paper cutter(destroyer)
	application	one massive object, that is dropped into a can	object rolling or slipping out of a can
multiple objects	documents	crumpling up of papers	paper cutter(destroyer)
	applications	several objects dropped in a can	multiple objects rolling and slipping out of a can
	mixed	several objects dropped in a can	multiple objects rolling and slipping out of a can, combined with the sound of a paper cutter

It is useful to make a coarse assortment of the possible states of the trash-can and the related actions upon it. This will facilitate the selection of proper sounds, for instance to indicate if the trash-can is empty. The following states for the trashcan can be found:

Status:	empty	empty	empty	empty	empty	filled	filled	filled	filled	filled
object:	paper	paper	applic	applic	mixed	paper	paper	applic	applic	mixed
number	single	multi	single	multi	multi	single	multi	single	multi	multi

The status indicates if the trash-can was empty or not, before an object is dropped into it. In this way, an obvious difference can be made in the 'dropping' sound that is used.

Sounds for representing user actions:
activation: an explicit ' double click' sound.
selecting: the sound of a (foot-) switch.
dragging: the sound of a lawn-mower.
resizing: the sound of adhesive tape being removed. The speed of the resizing action can control the produced speed of pulling of the tape. An alternative for resizing can be the sound of sucking air or liquid into a suction-pump.

Sounds for indicating the stack-level of windows or other objects:
For each stack level, a different formant structure for the sound can be used.
Each window at a certain stack-level can have formants in an spectral area, that is specifically assigned to represent that level. The total amount of levels that can be represented in this way is not endless, but it should be possible to design a great amount of combinations in formant structure, that are easily distinguishable.

[1] Although the original trashcan object can only be found in the Apple Macintosh desktops, similar objects with the same functionality, can be found in most other GUIs as well. In these other GUIs, the visual appearance is then often a *recycler* icon.

17.2.7 Suggestions for sound synthesis techniques :

Back-ground sounds (granular synthesis):
For sounds that should mostly appear in the back-ground, like the window-sounds, granular synthesis techniques could be used. With this technique, sounds that have a very natural quality can be obtained. Because of it's structure, the granular synthesis technique almost automatically produces natural and vivid sounds.

Physical modelling:
A synthesis technique that is becoming quite popular the last few years, is 'physical modelling' (PM). The technique is based upon complete physical descriptions of real-world objects and the forces that manipulate these objects. The simulated interaction between the object and the manipulator produces sound waves, which can be calculated. In the ideal situation, this technique can deliver sounds that are exactly as vivid, realistic and manipulable as sounds that are produced with real objects. If an interface should somehow be an reflection of the real world and give users a natural feeling of interacting with this world, PM-techniques can be considered for the sounds that are used. A disadvantage of the PM technique for auditory interfaces is that the variety of possible manipulations can produce timbres that differ significantly from eachother. Confusion about the identity of the sound source is thus very likely to occur. Summarized, the great flexibility of PM and the natural quality of the produced sounds are advantages that can be of use, but the same aspects that can cause problems for the identifyability of sounds, as well.

Finally, I suggest that for a practical implementations of the auditory interface system, audio-data reduction techniques are used. Although hardware, like memory storage becomes increasingly cheeper, this will reduce the memory requirements and thus the hardware costs for the end user. There should however be no noticeable decrease in the perceived audio quality, thus adequate techniques with not to drastical reduction rates are recommended. For instance a technique, like the PASC (III) coding technique (Philips) .

References :

(Ballas): Ballas J.A.(1994)
'Delivery of information through sound.'
SFI Studies in the science of complexity, Addison-Wesley Publishing Company

(Blattner1): Blattner M.M, Sumikawa D.A. & Greenberg R.M.(1989)
'Earcons and icons: their structure and common design principles'
HCI , 4 , p. 11 - 44

(Blattner2): Blattner M.M, Papp A.L., Glinert E.P.(1992-'94)
'Sonic enhancement of Two-Dimensional Graphic Displays'
SFI Studies in the science of complexity, Addison-Wesley Publishing Company

(Brewster et.al.): Brewster S.A, Wright P.C., Edwards A.D.N.(1992-'94)
'A Detailed Investigation into the Effectiveness of earcons.'
SFI Studies in the science of complexity, Addison-Wesley Publishing Company

(Bölke&Gorny): Bölke L. , Gorny P. (mai 1994)
'Auditory Direct Manipulation of Acoustical Objects by Blind Computer Users'
Internal Report from the Faculty of Informatics, University of Oldenburg.

(Crowder): Crowder R.G (1993)
'Auditory Memory'
Thinking in sound, the cognitive psychology of human audition,
Oxford University Press (Clarendon Press)

(DiGiano et.al.): DiGiano C.J. , Baecker R.M. (1992)
'Program auralisation: Sound enhancements to the programming environment'
Proceedings Graphics Interface '92 (p. 44-52)

(DiGiano et.al.): DiGiano C.J. , Baecker R.M. , Owen R.N.(1993)
'LogoMedia: A sound enhanced programming environment for monitoring program behaviour.'
INTERCHI'93, ACM Press, Addison Wesley Publishing, (from p301)

(Duncan):Duncan L. R. (1993)
'Sound&hearing, a conceptual introduction.'
Lawrence Erlbaum Associates Publishers.

(Edwards): Edwards A.D.N.(1989a)
'Soundtrack: an auditory interface for blind users.'
HCI, 4, 45-66

(Edwards2): Edwards A.D.N.(1989b)
'Modelling blind users' interactions with an auditory computer interface'
Int. Journal of Man-Machine Studies 30 (p. 574-589)

(Edwards et al): Edwards A.D.N , Mynatt E. (1993)
'Enabling technology for users with special needs'
INTERCHI '93

(Edworthly, Loxly and Dennis): Edworthly, Loxly, Dennis (1991)
'Improving auditory warning design: relationship between warning sound parameters and perceived urgency'
Human Factors '91 (p. 205-232)

(Fraisse):Fraisse P. (1993)
'Rhythm and Tempo'
Thinking in sound, the cognitive psychology of human audition,
Oxford University Press (Clarendon Press)

(Gaver(1989)): Gaver W.W. (1989)
'The SonicFinder: An interface that uses auditory icons.'
Human Computer Interaction 4 (from p1)

(Gaver1): Gaver W.W.(1992-'94)
'Using and creating auditory icons'
SFI Studies in the science of complexity, Addison-Wesley Publishing Company

(Gaver2): Gaver W.W.(1990)
'Auditory Icons in large scale collaborative environments.'
HCI, Interact '90, 735-740

(Gerrits et. al.): Gerrits A.H.J., Poll L.D.H. , Waterham R.P. (Jun1993)
'VISA-Comp object library: Software engineering document'
IPO Rapport no. 916

(Houtsma): Houtsma A.J.M., Rossing T.D., Wagenaars W.M. (1987)
'Auditory Demonstrations'
CD + Booklet, Institute For Perception Research

(Kramer2): Kramer G.(1994)
'Some organising principles for representing data with sound.'
SFI Studies in the science of complexity, Addison-Wesley Publishing Company

(Leimann/ Schulze): Leimann E. , Schulze H.H.(Oct 1994)
'Earcons&Icons: An experimental Study.'
Interact '95 (paper), Norwegian Computer Society.

(Ludwig, et al.(1990)): Ludwig L., Pincever N., Cohen M. (1990)
'Extending the notion of a window system to audio'
IEEE Computer 23; (p66-72)

(Mynatt): Mynatt E.D. (1992-'94)
'Auditory Presentation of Graphical User Interfaces'
SFI Studies in the science of complexity, Addison-Wesley Publishing Company

(Poll1 et. al.): Poll L.D.H., Beumer J.J., Waterham R.P. (Nov 1993)
'Graphical User Interfaces and Blind Users'
Rapport no. 943, Institute for Perception Research

(Poll2 et. al.): Poll L.D.H., Eggen J.H. (Mai 1994)
'GUI admission for VIPs: a sound initiative.'
Manuscript no. 1017, Institute for Perception Research

(Poll3): Poll L.D.H. (Jul 1994)
'MIDI software enginering document'
Institute for Perception Research

(Poll4): PollL.D.H, (beginning of 1995)
'Transparent GUI admission for non-sighted users:
exploring a non-visual interaction device'
IPO Manuscript in Preparation V1.2

(Portigal): Portigal S. (Jan 1994)
'Auralisation of document structure'
Thesis for the degree of Master of Science, University of Guelph

(Rösing): Rösing H. (1972)
'Die Bedeutung der Klangfarbe in traditioneller und elektronischer Musik.
Eine Sonographische untersuchung.'
Musik Verlag Emil Katzbichler München.

(Rossing): Rossing T.D.(1990)
'The Science of Sound.'
Addison-Wesley Publishing Company.

(Scaletti): Scaletti C. (1994)
Sound synthesis algorithms for Auditory Data Representations
SFI Studies in the science of complexity, Addison-Wesley Publishing Company

(Temp): Tempelaars S., (1992)
Taken from notes, that were made during a coarce on 'aspects of tone sensation'
during season 1993/ 1994.

(Thimb): Timbleby H. (1992)
'Human Computer Interfaces'
Cambridge University Press.

(Versfeld): Versfeld N.J.(1992)
'On the auditory discrimination of spectral shape.'
Proefschrift IPO (p. 69- 86).

(Wenzel et.al):

Wenzel E.M., Foster S., Wightman F.L., Kistler D. J. (1989)
'Realtime digital synthesis of localized auditory cues over headphones'.
NASA rapport 1989

Begault D.R. ,Wenzel E.M. (Aug1990)
'Techniques and applications for binaural sound manipulation in human machine interfaces.' NASA rapport Aug1990

Begault D.R. , Wenzel E.M. (Oct1990)
'Technical aspects of a demonstration tape for three dimensional sound displays.'
NASA rapport Oct1990

Wenzel E.M. , Wightman F.L. , Kistler D. J. (mid-1991)
'Localization with non individualised virtual acoustic display cues'

Wenzel E.M., Wightman F.L. , Kistler D. J. (Sept 1991)
'Headphone localization of speech stimuli.'
NASA rapport Sept1991

Wenzel E.M., Wightman F.L. , Kistler D. J. (Oct 1991)
'Three dimensional virtual acoustic displays.'
NASA rapport Oct 1991

Wenzel E.M., TaylorR.M., Foster S.H. (Oct 20 1991)
'Real-time synthesis of complex acoustic environments.'

(Williams): Williams S.M. (1994)
'Perceptual principals in Sound Grouping.'
SFI Studies in the science of complexity, Addison-Wesley Publishing Company

(Yost): Yost W.A. (1994)
'Fundamentals of hearing, an introduction,3rd edition'
Academic Press Inc

(Zwicker): Zwicker E. ,Fastl H. (1990)
'Psycho-acoustics, Facts&Models'
Springer Verlag Heidelberg

AppendixA The exact details of some sound suggestions and guidelines:

In this paper, guidelines for the creation of sound sets and suggestions for further extension of an auditory interface have been provided. Most of the descriptions don't go very much into detail, but give an indication of the possibilities and troubles with respect to sound. Nonetheless, a more detailed and specific description of the techniques that can be used and their relationship to the physical properties of a GUI is necessary for the actual implementation. This chapter provides for such a closer look.[1]

A.1 Amplitude, the relationship between decibels, MIDI volume and distances in screen pixels:

In the guidelines for auditory interface design, an amplitude range of 20 [dB] was suggested, starting from ± 10 [dB] above threshold of hearing. These guidelines should somehow relate to the physical properties of a GUI. This is necessarily for putting up rules, that enable the translation of distance information between objects on the screen into MIDI-control parameters. To form such rules, the properties of the three dimensions; GUI, MIDI and the physical parameters of sound, have to be compared.

Comparing the resolution and the range of these three dimensions:
- A relevant stepsize for amplitude changes in the physical dimension of sound is ±1[dB]. The proposed amplitude range, covers 10 - 30 [dB].
- The screen resolution of a GUI is, 480*640 [pixels] when a standard VGA screen is used.
- The controller values that apply for MIDI mostly range from 0 - 127.

When the parameter ranges and the resolutions of these three dimension are combined, some useful relationships can be extracted. These relationships are:

The largest possible straight distance on a 14' VGA screen is, diagonal, 800 [pixels]. This value can be divided by the range of 127 MIDI-controller-values. A step of 1 controller value will then correspond to a step of 6.3 [pixels] in the GUI.
If the 127 controller values should coincide with an amplitude range of 20 [dB], a step of 1 controller value will correspond to a step of 0.16 [dB] in amplitude. However, it makes more sense to choose a stepsize of at least 1 [dB]. Such a stepsize is represented by a MIDI step of 6.35 controller values. Relating these sizes to the physical properties of the GUI screen, means that every step of 6.35*6.3≈ 40 [pixels], will result in a change in amplitude of ± 1 [dB].

If this relationships is straightforward implemented, the following formula applies:

$$\boxed{\Delta Ctrl = \left|\frac{\text{distance}[pix]}{\text{stepsize}[pix]}\right|} \qquad \text{(F A.1)}$$

This formula describes the difference expressed in MIDI controller value (Ctrl), that represents the difference in amplitude, between objects that are at a certain distance to eachother. When the stepsize of 40 [pixels] is used, the total amount of possible control steps is: 16 horizontally, 12 vertically and 20 diagonally.

Unfortunately, for MIDI controller values, only integer values can be used. Thus a stepsize of 6.35 in controller value is not possible. In practice a stepsize of 6 MIDI-volume controller values is therefore recommended. This value will corresponds to a stepsize on the GUI-screen of 37.8 [pixels]. The maximum, total amount of possible control steps will then change to 21 diagonally.

[1] Formulas that are mentioned in this chapter, but not further explained, can be found in appendix B.

A. 2 Duration compensation, by changing the MIDI-volume of sounds: In relation to the effects of temporal integration on amplitude, durations shorter than 150 [ms] should be amplitude-compensated. Below this 150 [ms], it would be sufficient to introduce +0.09 [dB] compensation for each 1 [ms] that the duration becomes shorter. This value is quite close to twice the amplitude stepsize, that each MIDI-volume ctrl value represents. To estimate a duration compensation curve, an increase in MIDI-volume ctrl of 1 value could be used to compensate for every shortening of the duration by 2 [ms].In most cases this will be sufficient to keep short sounds, with a varying duration, properly balanced.

A. 3 Representation of stacked windows by band filtered, white noise: The auditory representation of windows, can be done with band-pass filtered noise. For each stack-level a specific frequency band can then be reserved. For the actual implementation, several approaches were considered.

a)A first approach was to divide the complete frequency range in frequency bands of equal width. The number of bands could than be related to the possible, total amount of distinguishable steps in amplitude. For the discussion on this topic, I refer A.3.1. This approach, will result in a grid of 21 frequency bands. Each band will have a bandwidth of 951.5[Hz]. Unfortunately, in such a linear model, the properties of the human hearing system are not taken into account. This can lead to problems, in that:

- a bandwidth Of 20 [Hz] - 20 [kHz] is to broad (most people won't be able to hear the highest and lowest bands).
- The ear doesn't perceive frequency relationships on a linear scale, so some bands may be to small to be independently perceived, while others are to wide.

b) A second approach was based on the results of experiments on the detectability of changes in spectral slope (Versfeld). These experiments indicated that a change in a spectral slope is very well audible, when signal bandwidths of 3 - 6 semi tones are used. With this fact in mind, a bandwidth of 6 [ST] could be chosen, for representing each stack-level of the windows. Again, the grid will then consist of ± 21 bands, however with units in semi tones.As a result, the scale is here logarithmical, instead of linear. Although this approach seems to correspond better to the way in which we perceive, a drawback is that it only relates to hearing differences in spectral slope, within a given frequency band. Nothing is said about the perceivability of transitions and differences between different frequency bands.

c)The third and final approach is to relate the used frequency bands to the critical bandwidths of our hearing system. The critical bandwidth is ± 90 [Hz] for frequencies up to ± 600 [Hz]. For higher frequencies it is 15% of the centerfrequency .
According to the '15%-rule', the critical bandwidth is 90 [Hz] at 600 [Hz]. , Therefore it is best to choose this frequency value as transition point, between the two methods for determining the critical bandwidth. The bandwidth then changes from a constant 90 [Hz] for frequencies below-, to 15% of the centre frequency, for frequencies above this value.

The MIDI-note, that corresponds best to this 600 [Hz] value, can be found with formula B3. The result indicates that the proper note can be found 5.4 semi tone steps above A4 = 440 [Hz]. To ensure that every used frequency band, generated with the '15%-rule', will exceed the minimum of 90 [Hz], this value can be rounded to 6 [ST].

Going up 6 [ST] from A4, adds up to d#5 . This pitch has a frequency of 622,25 [Hz].

Below d#5, formula B4 applies, for calculating the bandwidth in semi-tones, that best matches the laid down 90 [Hz] value.

Above d#5, the bandwidth is 0.15 / 0.058 = 2.59 [ST]. This is the minimum value, necessary, to ensure of the individual detectability of the bands.This value can best be rounded to 3 [ST]. All the successive centre frequencies should thus be spaced 3 [ST] apart, to avoid that used bands will overlap.

Figure A.3.1 shows an overview of the widths of the critical bands, from very low, to mid-centre-frequency ranges.Indicated are key-names,frequency and STmean values[1].

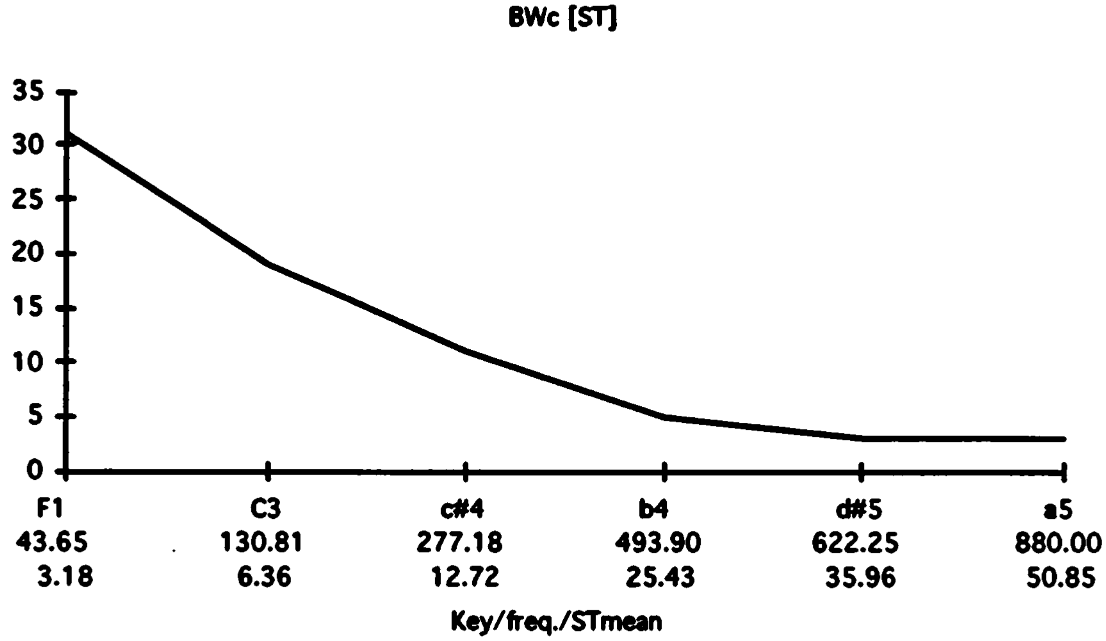


Fig. A.1

When one counts through, from d#5, in steps of 3 [ST], the number of frequency bands that can be represented above d#5 will add up to 17. Below d#5 only 5 frequency bands are available for the representation. If fixed centre frequencies are used, the total number of critical bandwidths will thus theoretically add up to 22.
This is only true, when the complete audible frequency range is used. It still has to be seen if the very low and high frequency ranges can be used for the auditory interface.

A. .4 Equal-loudness-curve compensation:

With the division in critical bands, for the representation of windows, another problem arises: the relative working of our hearing system, as far as loudness judgements are concerned. The selected amplitude-range was 10 to 30 [dB] above threshold of hearing, which depends on the amount of masking back-ground noise.
Most sounds will thus be presented at a level of ±20 [dB]. The 20 [phone]-curve is therefore most useful to serve as reference for a simulation of loudness-curve compensation.

This curve is approximately flat in the frequency range of ±500-2000 [Hz], corresponding with a key-range of b4 to b6 (Formula B3). Above b6 and beneath b4, loudness compensation is necessarily, to ensure that sounds are being perceived equally loud, independent of the frequency range in which they are played.

If the lowest key to be represented, is related to the bandwidth of our hearing, this key should correspond to a frequency value of ±20 [Hz]. Formula B3 shows that this is the key d#0. However, in the guidelines a minimum frequency of 125 to 150 [Hz] was suggested. When ±150 [Hz] is chosen to be the lower border of the used frequency range, this will correspond to key d3 (147 [Hz]). The suggestion for the upper boundary of the frequency range was ±5000 [Hz], which corresponds to key d#8 (4978 [Hz]). These guidelines are however related to pitched sounds.
The spectral contents of the sounds may exceed these boundaries, as long as the fundamental, if there is one, stays within the suggested range.
Note that noise bands, as used for the window representation, are quite broad in the higher frequency ranges, e.g. 866.2 [Hz] around d#8. For this reason and because it should be possible to use non-pitched sounds as well, the frequency ranges exceeding the boundaries must also be amplitude compensated.

For amplitude compensation, the entire key-range was divided into 5 segments, with each their own amplitude curve. These curves estimates as good as possible, the shape of the original phone curve.

A first estimation-approach was by using linear curves. This did not work out well, because the influence of the curve increased to quickly, for every semi-tone step up- or downwards. This meant that the values at the beginning and the end of each linear curve fitted fairly well, but the values in between diverged to much of the original phone curve.

A second attempt was made, using low-pass filters. The curves that these filters show are more smooth than the linear approximations. Unfortunately the relative attenuation, provided by these filters was not enough, when compared to the actual phone-curves. A second effect was that the low frequencies in every frequency band were favoured as compared to the higher frequencies, due to the spectral shape of the filter curves.

A third way to balance the loudness over the frequency range, is purely by hearing. However this approach is too insecure and depends very much on the timbre of the sound that is being used and on the subject, that judges the variations in perceived loudness.Further more, this method is very time consuming, because each specific sound has to be separately balanced over the entire frequency range.

The last approach, that was finally implemented, depended on some very specific functions of the Kurzweil K2000 synthesizer that was used. Functions, of which the outputs depended on specific key-numbers and key-ranges, and quadratic equations were combined. As a result, the amplitude ratios were shaped in such a way, that the resulting amplitude versus frequency curve, estimated the original phone-curve quite well. The slope of the segments of this implementation resembled much of quadratic curves. Figure A.2 illustrates the implementation.

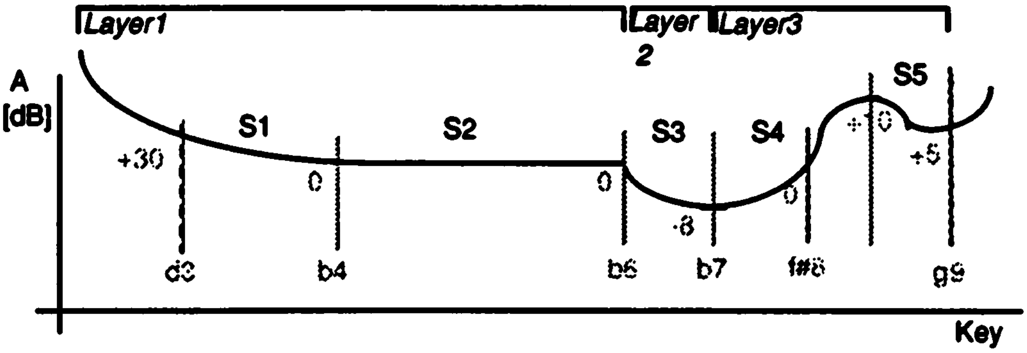


Fig. A.2
The 'layers' displayed here, indicate the key-ranges, that were covered by each sound-layer of the synthesizer program. The decimal values indicate the amount of attenuation, at the specific key that is displayed below each value. With this final approach satisfying results were obtained with the windows sounds. Most probably it will work fine for most other sound, like the Auditory Guidance, as well.

A. 5 Doppler effects:

Doppler effects can be introduced to underline the occurrences of movements within the GUI. In this way the user obtains a more natural impression of actually moving through the interface with the mouse pointer, when the Auditory Guidance is being used. Actions like dragging, or the occurrence of other object movements, could be provided with a Doppler effect as well. For the practical implementation of this effect, a link between the speed of the movements of the user through the interface and the amount of effect, that will be the result of those movements is necessary.
In chapter 6, under 'Timing', it was found that 100 [pix/ sec], could be used as an indication for the maximum speed value. If one defines that 1 pixel, corresponds to 1 decimeter in real physical units, the maximum speed in pixels will correspond to a maximum speed of 10 [m/sec] physically.

The amount of frequency adjustment, resulting for a receiver moving towards or away from a fixed object, can be calculated with:

$$f' = f \cdot \left(\frac{c}{c + r.v_S} \right) \Leftrightarrow \frac{f'}{f} = \left(\frac{c}{c + r.v_S} \right)$$

c = speed of sound, here set at 343 [m / s] (F A.3)

vs = unit vector, indicating if the receiver is moving towards (-1),

or away from the object (+1).

At the maximum speed of 10 [m/s], the resulting frequency will be 1.03003 * f , for a receiver moving towards the object. Especially when the effect is applied to the Auditory Guidance, it is important to look at the ratio's between the resulting frequency deviations and the Just Noticeable Differences in pitch, that are in affect at these frequencies. Suppose that the pitch range used for the Auditory Guidance is d3 to d8, than the following relations can be found:

The Just Noticeable Difference in frequency is defined as 1/30 of the critical bandwidth at a certain frequency. In this way the JND$_f$ will be 1/10 semi-tone for frequencies above d#5 and 1/30 (2.5 + 0.614 * ΔST) for frequencies below this key. If the resulting frequency and the frequency-difference, at maximum speed, are compared with the JND$_f$, the following values will show:

Key	f' [Hz]	Δf [Hz]	JND$_f$ [ST] & [Hz]	ST$_{mean}$[Hz]	JND$_f$ < Δf < ST$_{mean}$
d3	151.24	4.41	0.47 [ST] = 4.0[Hz]	8.5	4.0< 4.409 < 8.5
a4	453.21	13.21	0.207 [ST] = 5.3[Hz]	25.52	5.3< 13.21 < 25.52
d8	4873.6	142.08	0.1 [ST] = 27.4 [Hz]	274.42	27.4<142.08< 274.42

Table A.1

As these values show, the frequency deviations that are introduced by movements at maximum speed , are always perceptible. Still, these deviations stay well under the size of one semi-tone step at the same frequency. In this way, they won't interfere with the pitches that are used by the Guidance to indicate the positions of objects.

A. 6 Panning, according to the horizontal position of the objects:

In connection to the topic of representing positions in the auditory domain, panning is probably the most simple issue. Depending on the number of panning steps the synthesis device can handle and the screen resolution of the GUI, a simple relation between these two dimensions can be set up. The panning resolution of MIDI devices is often much less than 127 steps. The screen resolution was set at 640×480, until now.

The total amount of horizontally displayed pixels can be divided by the number of possible panning steps. The result is a stepsize of one controller value per XX pixels.
If the common MIDI-resolution of 127 steps, is chosen as reference, each movement over 5 [pixels], should result in an in-, or decrease of one panning-controller value.

For the actual implementation, some specific panning functions of the K2000 were used. The range of 127 was resealed into values ranging from -100 to +100% .
These settings then corresponded to panning positions entirely left (-100%) and entirely right (+100%). The reason for using these functions is that they provided a more smooth panning progress, than the conventional MIDI-panning control.

A last remark related to panning is that sine/ cosine curves are recommended for controlling the amplitude ratio's between the left and right channel. With such curves, the total loudness of a sound will stay constant, regardless of the panning position it has. With linear curves this cannot be obtained and the result will be an amplification of + 6 [dB] in the centre position, compared to the levels at the far left or right positions.

Appendix B. translation formulas:

For the implementation on MIDI-based equipment, formulas to retrieve the proper parameter settings for this equipment, are necessarily. Therefore the following formulas have been used:

1 Mean stepsize of one semi-tone (ST):

At a certain frequency, the mean extent of one semi-tone step is the average of going one semi-tone up, or down from this given frequency. Thus the mean ST-stepsize will be:

$$\boxed{\mathrm{ST}_{\mathrm{mean}} = \frac{\sqrt[12]{2}\cdot f_C - \left(\sqrt[12]{2}\right)^{-1}\cdot f_C}{2} \approx 0{,}0577946\cdot f_C \approx 0{,}058\cdot f_C} \qquad \text{F B1}$$

2 Frequency indications in Herz, derived from intervals in semitones:

When a distance 'n' to A4 is given in semi-tones, the frequency of the pitch at that distance can be calculated in Herz by:

$$\boxed{f_C = 440\cdot\left(\sqrt[12]{2}\right)^{n}} \qquad \text{F B2}$$

3 Interval distance 'n' to A4, of a frequency given in Herz:

If f_C is the given variable, the distance 'n' to A4 in semi-tones can be calculated by:

$$\boxed{\mathrm{ST}_{\mathrm{distance}}:\ n = \frac{\ln f - \ln 440}{\frac{1}{12}\cdot\ln 2}} \qquad \text{F B3}$$

E.g. suppose one searches the interval distance to A4 for a tone of 1000 [Hz].
'n' will then be 14.2. If the corresponding key is being searched for, one can find this key by going up 14 + 1/5 semitones from A4. This results in B5 + 1/5 [ST_{B5}] ≈ 999,51 [Hz].

When this should be introduced on the MIDI equipment, the key-number of B5 will represent the coarse tuning, while the adjustment of 1/5 [ST], is accomplished by fine tuning the pitch parameters.

4 Critical bandwidth, indicated in semi-tones:

If the critical bandwidth should be expressed in semi-tones, a distinction has to be made in the critical bandwidth above and below key d#5. The reason for this will be explained in the discussion of the noise-bands for representing window-stack-levels.
Above d#5 the critical bandwidth can be represented by 3 [ST] bands.
Below this key the critical bandwidth, 'BW_C ', in mean semi-tones, can be calculated with:

$$\boxed{\mathrm{BW}_C = (2{,}5 + 0{,}614\cdot\Delta ST)} \qquad \text{F B4}$$

ΔST is the interval distance, below d#5, given in semi-tones. Thus ΔST is:

$$\boxed{\begin{array}{l}\Delta ST = (\text{keynumber of d\#5}) - (\text{keynumber of target frequency})\\ \Delta ST = 75 - \mathit{keynumber}\end{array}} \qquad \text{F B5}$$

When not a key, but the critical bandwidth itself is the given variable, the KeyNumber, corresponding to the frequency that belongs to this critical bandwidth, can be calculated:

$$\boxed{\Delta ST = \frac{BW_c - 2.5}{0.614}, \qquad KeyNumber = 75 - \frac{BW_c - 2.5}{0.614}} \qquad \text{F B6}$$

Appendix C. MIDI Control Sources and Sequences:

This appendix contains the listings of the MIDI-sequences, needed for playing the audicons and earcons. In these sequences, controller data is also occassionally used. Therefore an overview of the reserved controllers, will be provided at first.This overview includes a short description of the effect that these controllers most often elicit. Of coarse, the effect depends on the way each program on the Kurzweil is structured. Therefore, the effects are not always exactly the same for each program, but an effort was made to be as consistent as possible. The assignment of control sources was:

Ctrl-n	Ctrl-name	Source	ctrl-value	
114	LFO1	Pitch-Src1	127 / 0 (OFF)	Causes vibrato-effect when ON
92	TremoloDepth	Amp-Src2	127 / 0 (OFF)	Causes tremolo-effect when ON
19	ControlD	Position-Src	0 - 127	Regulates panning, center is 64
16	ControlA	Cut-off, Amp-Src1, Shaper	127 / 0 (OFF)	Controls the explicit spectral changes, that are used as cue. To compensate for deviations in amplitude, resulting from those changes, simultaneously the amplitude of the sound is controlled by the same ctrl.
17	ControlB	BandWdth-Src1		Controls the bandwidth, when a structure with bandpass filters is used.
18	ControlC	Alt-Switch	127 / 0 (OFF)	
106	VTRIG1	ASR1	111 (ff , on)	
107	VTRIG2 (Rev)	ASR2	31 (pp, off)	

Table C.1

The first three controllers apply to all programs. One of these controllers will only be omitted in a sound program, when it is not used. Note that,e.g. panning will not be used for continous, stereo sounds. However, all programs have in principle control number 19 set for panning. An initialisation of this controller, before the sound is being played, is therefore always necessary. Otherwise the panning will be set according to a default value. This is controller number 0, which results in the sound being panned entirely to the left. Before a note is triggered, a value of 64, or a value that corresponds to the position of the represented object on the screen, should be send for panning control. The sound will then appear in the centre, or at a panning position that relates to the horizontal screen-position.

In the sequence listing, all default controller messages, like panning control, should be send before the actual note-on message. This to ensure that each parameter is set to it's proper value. The order of the MIDI-messages in the sequence list is:

Header:	Comment:
MIDI-Channel	Default channel is 1, but when this channel is already occupied by a sound, the first available channel should be choosen.
Program-Change-Number	See discussion, under 'notes related to the use of the K2000'.
Panning-Controller-Number + Data	Default value is 64 ('DEF'). If also variable panning is possible, this will be indicated by the word 'VAR'.
Optionally:	
Controller-Number +Data	The first one of the two optional, controller messages that should sometimes be included in the trigger sequens.
Controller-Number +Data	A second optional controller, necessarily for some sound programs that make use of two external control sources.
Key-Triggering:	
KeyON: KeyNumber+Velocity	The default velocity value is 110, corresponding to an amplitude of ± 26 [dB] above hearing treshold).

Table C.2

In the first table, all the objects are listed with a fixed state. Explicit state changes that can occur to objects are seperately listed in the third list. This, because these state changes consist of combinations of the fixed states as listed in table 1, that evolve in time.

TableC3: Objects, states and sequences:

Object:	State	CH	PRG	PAN	CTRL,v	CTRL,v	Key	Vel
Button	Default	1	1	Def/Var	16, 0	-	24+21	110
		"	"	"		-	24	"
	Non-Selectabl	"	"	"	16, 127	-	24	"
CheckBox	Unmarked	"	1	"	16, 0	-	26	"
	Def-Unmarked	"	"	"	16, 0	-	27	"
	Marked	"	"	"	16, 0	-	26+28	"
	Def-Marked	"	"	"	16, 0	-	27+28	"
	Non-Selectable	"	"	"	16, 127	-	26	"
RadioButto	Off	"	1	"	16, 0	17, 0	29	"
	ON	"	"	"	16, 0	17, 127	29	"
	Def-On	"	"	"	16, 0	17, 127	29+41	"
	Non-Selectable	"	"	"	16, 127	17, 0	29	"
EditCntrl	Active	"		Def	16, 0	-	48	"
	Non-Active	"		Def	16, 127	-	48	"
Edit-line	Hit/ Select	"		Def/Var	-	-	50	"
Text-line	Hit/ Select	"		"	-	-	50	"
List-Box	Above	"		"	-	-	52	"
ListasPD	Appear	"		"	-	-	53	"
	Disapear	"		"	-	-	55	"
(List)	Scrolling	"		"	?	?	57	"
	Stop/ Select	"		"	-	-	59	"
MenuList	Appear	"		"	-	-	62	"
	Disappear	"		"	-	-	64	"
(HorMList	Above	"		"	?	?	65	"
MenuItem	Non-Highlight	"		"	16, 0	18, 127	67	"
	Default (highl)	"		"	16, 0	18, 127	67+69	"
	Non-Selectable	"		"	16, 127	18, 127	67	"
PD-edit	Non-Highlight	"		Def	16, 127	-	71	127
	Default (highl)	"		Def	16, 0	-	71	15
Icon	Non-Highlight	"		Def/Var	16, 127	-	60	110
	Default (highl)	"		"	16, 0	-	60	"
DialogueBo	Appear	"		Def	-	-	96	"
	Above	"		Def	-	-	97	"
	Disappear	"		Def	-	-	98	"
Graphic	Above	"		Def	-	-	72	"
Pointer	Busy	"		Def/Var	-	-	74	"
Hit1	Hit General	"		"	-	-	84	"
Hit2	Hit Window	"		"	-	-	85	"
Select1	Select cursor	"		"	-	-	86	"
Select2	Select default	"		"	-	-	87	"
Dragging		"		Var	?position	?speed	88	"

The Auditory Guidance doesn't have a specific key-number for it's activation, because it covers an entire key-range. Therefore, the Guidance program should be played at a seperate MIDI-channel. It also will have a variable panning setting, (see Guidance design ideas, appendixA). The state of the Guidance is simply either ON or Off.

For the window sound, a key-range from f1 untill d#8 is reserved. It is recommended to assign a fixed MIDI-channel to the window sound, which is also a usefull thing to do for the Auditory Guidance. The actual key-numbers, representing the corresponding stack-levels of the windows are listed here next.

Table C4: the window object:

Object:	State	CH	PRG	PAN	CTRL,v	CTRL,v	Key	Vel
Window	appear		7	Def	-	-	111	15
	disappear		"	"	-	-	Variable	127
	above (active)		"	"	-	-	111	110
	above (other)		"	"	-	-	108	"
	"		"	"	-	-	105	"
	"		"	"	-	-	102	"
	"		"	"	-	-	99	"
	"		"	"	-	-	96	"
	"		"	"	-	-	93	"
	"		"	"	-	-	90	"
	"		"	"	-	-	87	"
	"		"	"	-	-	84	"
	"		"	"	-	-	81	"
	"		"	"	-	-	78	"
	"		"	"	-	-	75	"
	"		"	"	-	-	71	"
	"		"	"	-	-	61	"
	"		"	"	-	-	48	"
	"		"	"	-	-	29	"

As announced, table5 contains the sequences for the possible state changes that can occur.The duration of those changes is mostly 250 [ms], with a maximum of 380 [ms] for combined changes. Combined changes occur, when one state change automatically causes another state change to happen. Both state changes will then be represented , which results in a somewhat longer sequence. Such sequences are not twice as long, because the second state change already starts, when the first one has not entirely finished yet. The time that is covered by the sequences is subdivided into segments. The segments are indicated in the table as a, b, c and d. For the non-combined sequences, only a and b are of importance. The old state is presented during the a-segment and the new state starts with segment b. Segment c, covers the entire 250 [ms], for the presentation of the entire state change. The c segment is used, when a sound already represents a compltete state change by itself. E.g. the sound that represents the appearance of a pull-down list. Segment d is only used for combined state changes. In this segment the new state of the second state change will start. Sometimes, key numbers should not be turned of at the end of the segment to which they belong, but they should be sustained into the next segment. This will be indicated by keynumbers that are between parenthesis. A simple figure of the segments and their durations:

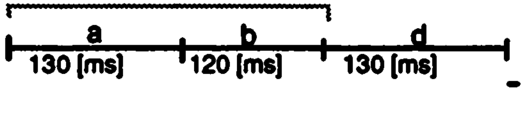


Fig C.1

TableC5: State changes:

Object:	State Change	Seq	CH	PRG	PAN	CTRL,v	CTRL,v	Key	Vel
Button	X-> default	a	1	1	Def/Var	16, 0	-	24	110
		b	"	1	"	"	-	(24)+21	"
CheckBox	X->Def-Marked	a	1	1	"	16, 0	-	26	"
		b	"	1	"	"	-	27+28	"
	X->Default	a	1	1	"	16, 0	-	26	"

		b	"	1	"	"	-	27	"
RadioButton	Def-On	a	1	1	"	16, 0	17, 0	29	"
		b	"	1	"	"	17, 127	(29)+41	"
ListasPD	Appear	c	1	2	"	-	-	53	"
	Disapear+ highlight PD	c b d	1 2 "	2 3	"	- 16, 127 16, 0	- - -	55 71 (71)	" 127 15
MenuList	Appear+ highlight MenuItem	c b d	1 2 "	3 3	"	- 16,0 16,0	- 18, 127 18, 127	62 67 (67)+69	110 " "
	Disappear	c	1	3	"	-	-	64	"
MenuItem	X->Highlight	a	"	3	"	16, 0	18, 127	67	"
		b	"	3	"	"	"	(67)+69	"
PD-edit	X->Highlighted	a	"	3	Def	16, 127	-	71	127
		b	"	3	Def	16, 0	-	(71) .	15
Icon	X->Highlighted	a	"	3	"	16, 127	-	60	110
		b	"	3	"	16, 0	-	(60)	"
DialogueBox	Appear	c	"	4	Def	-	-	96	"
	Disappear	c	"	4	Def	-	-	98	"
Pointer	Busy	•1	"	5	Def/Var	-	-	74	"
Select1	Select cursor	c	"	6	"	-	-	86	"
Select2	Select default	c	"	6	"	-	-	87	"
Dragging		* 1	"	6	Var	?position	?speed	88	"
Window	Appear	c	"	7	Def	-	-	111	15
	Disappear	c	"	7	Def	-	-	111	127
	X->Active	a	"	7	Def/Var	-	-	<111	110
		b	"	7	"	-	-	111	110

Some important notes related to the use of the Kurzweil K2000 synthesizer:

The programs of the audicon sound sets in the Kurzweil, are numbered according to the classification into families. In this way, the objects shared under family A can be found under program change number 1, those of family B under number 2, etc.
These program change numbers are however relative. Number 1 only means the first entry the currently selected 'bank'. When absolute program change numbers are used, the bank numbers also have to be included. A problem occuring then, is that it could be that the sets are not always loaded into the same bank. As a partial solution to this problem the programs have been stored into 'Quick Acces Banks' (QA). Here, program change number 1, always means the first entry in the current QA-Bank, regardless of the absolute program change number that is attached to it. To use the set-up in this way, the user has to make sure that the following parameters are set properly:

- The program change setting at the MIDI-RECEIVE page should be set to QA-0-127,
- the Kurzweil should be in Quick Acces Mode and
- the proper QA-Bank has to be selected. It should be easy to find this bank, because it has the name "XX".
- Before starting of, the QA-entry 10 can be triggered, which contains a 1000 [Hz], calibration tone program at -30 [dB]. The output level should be adjusted untill this tone is just noticable.

For playing back the earcon sequences, the tempo has to be set at 120 [Bpm], according to the guidelines for the creation of earcon sets. Each earcon sequens has it's own sound program and thus its own progream change number. These programs are not stored into a QA-Bank, because such a bank has only got 10 entries. Extra attention has to be paid to, into which bank the the earcon programs are loaded.