# PATTERNS IN AUDITORY MENU DESIGN

*C. Frauenberger, T. Stockman*

Interaction Media Communication Group, Department of Computer Science,
Queen Mary University of London

frauenberger@dcs.qmul.ac.uk, tonys@dcs.qmul.ac.uk

## ABSTRACT

The design of auditory displays suffers from the lack of re-usable design knowledge, leading to ad-hoc solutions and inappropriate use of sound in human-computer interaction. We propose to tackle this problem by employing design patterns to capture design knowledge and make it available for designers of auditory displays to share and re-use solutions. In this paper we describe how we designed auditory menus by using design patterns which were developed for a prior prototype and refined according to the results of the prior evaluation test. The resulting auditory display employs 3D virtual audio environments with concurrent audio streams and was tested against state-of-the-art screenreader technology. The evaluation showed that flaws identified in the prior prototype were eliminated, but despite the improved naturalness the performance was only marginally better than with the screenreader. The patterns originated from graphical pattern sets and the auditory design is still an ad-hoc solution. The requirements for a general framework to capture good practice, coding of valid design knowledge and applying patterns to design problems are discussed.

## 1. INTRODUCTION

In State-of-the-Art user interface design the use of audio as an interaction modality has low priority. For example, the Apple Human Interface Guidelines [1], being the basis for many applications meeting the highest requirements in usability, mention the use of sound only as assistive technology for Apple's screenreader. This is in spite of the fact that the first experiments to create auditory displays for existing computer applications were undertaken in the mid 80s. Gavers SonicFinder intended to create an auditory display for the file manager application Finder in Apple operating systems [2]. However, our knowledge about using sound to effectively communicate interactive information, whether as a replacement for visual displays, or to provide a complementary modality, is still severely lacking compared with the development of visual displays. There is a need for a solid methodological basis which designers could use systematically to produce effective auditory displays and allow audio to play a more substantial role in human-computer interaction.

The benefits of having a robust auditory communication channel in the human-technology interface are striking. As well as reducing the bandwidth requirements on the often overloaded visual channel, the increased mobility of the user demand a shift towards interface technologies that do not consume much physical space [3]. Sound is a very good candidate for use in such devices (see also [4]) and many specific solutions have shown that it can be used as a highly efficient interaction channel (e.g. [5, 6]). However, *"... the lack of design guidelines, common for the creation of graphical interfaces has plagued interface designers who want to effectively build on previous research in auditory interfaces"* [7]. Previous research that addresses this problem includes the synthesis of organising principles by linking perceptual issues with practical implementation [8], the definition of an auditory design space similar to visualisation design spaces [9], the creation of sonification patterns [10, 11] and the specification of a design process for auditory displays [12]. However, principles and guidelines in auditory display design remain underused, often leading to ad-hoc solutions and inappropriate use of sound in user interfaces [13].

In order to make auditory display design knowledge more widely available, it is essential to capture design knowledge from existing work and bring it into a form that is easily accessible and applicable to design problems at hand. This implies various difficulties that we aim to address in our research: first, the elicitation of design knowledge from previous research. Little work in the field provides the rationale for design decisions, this is usually implicit and driven by the designer's experience. Second, the evaluation of the elicited design knowledge. Common practice must be backed by research to become valid and good practice. Third, the applicability of the design knowledge. To support designers in the process of creating auditory displays the knowledge provided must be invariant of the specific context of use. It must be applicable to different problems without decreasing quality. Finally, the extensibility of the design knowledge. In a growing community it is essential to make methods available to extend and refine the commonly shared design knowledge.

This paper advances the idea of capturing design knowledge for auditory displays through design patterns [14]. We believe that this is a very promising approach to promote the design of auditory displays as a discipline for a number of reasons, which we will elaborate in the next section. Subsequently, we will go into further detail regarding the design of specific patterns in order to show the development process. In the following three sections the definition, auditory implementation and evaluation of menu patterns is illustrated, before providing a summary of the paper and giving an outlook on future research.

## 2. PATTERN DESIGN

### 2.1. History

Design patterns originate from the discipline of Architecture where Alexander first developed patterns for buildings and urban development [15, 16]. His idea of patterns was to capture the *"quality without a name"*, i.e. the core of a solution to recurring design problems. Further, he aimed at providing all stakeholders in architecture (such as the people commissioning the work, the inhabitants and the architects) with a common language in order to communicate their ideas and requirements. More recently, design patterns became very popular in the field of object-orientated programming [17]. Although the approach proved very effective in enabling the re-use of design knowledge, it was not the same concept that Alexander had in mind. In contrast to Alexander's patterns, these were reduced to provide templates for code and have little expressive power for other stakeholders.

Design patterns have been applied successfully to various fields. In human-computer interaction pattern design has recently received increased attention and pattern sets have been published for various application domains. The design of interactive web content, for example, has shown that there are many recurring design tasks that can be described through patterns to provide other designers with guidelines [18]. There are also pattern sets for common graphical user interfaces or specialised sets for mobile devices (e.g. [19]) and the design of museum exhibits [20]. In all these applications design patterns were used to capture design knowledge, make it easily accessible to designers, enable its reuse and provide a common language for all stakeholders involved.

We believe that patterns are an important concept for a young design community such as the one for auditory display. It helps us to avoid the problem of re-inventing the wheel with every prototype. Design patterns provide a mechanism to develop and share our knowledge, often within the interdisciplinary groups which are the norm in auditory display design.

### 2.2. Design patterns for auditory displays

A major problem in auditory display design is that designers in general are strongly biased towards visual concepts when finding auditory replacements for existing visual displays. This often results in design flaws, as the two modalities have fundamentally different characteristics with widely differing strengths and weaknesses (e.g. see discussion about objecthood in [21]).

Tackling this problem, we propose to use user tasks and mode independent abstractions of user interfaces as the starting point for our design patterns. This approach allows for designing user interfaces at a level at which their mea ns of realisation is not yet determined and therefore not biased towards solutions in a specific domain. This also means that the instantiation of an interface in a specific modality can exploit the characteristics of the target modality. These mode independent design patterns - mi::patterns - therefor consist of an abstract description of the interaction problem, an abstract, mode invariant solution and guidance for instantiating the pattern in a specific modality. While we are focused on audio, this is also a possible process for incorporating other modalities. The concept is illustrated in figure 1

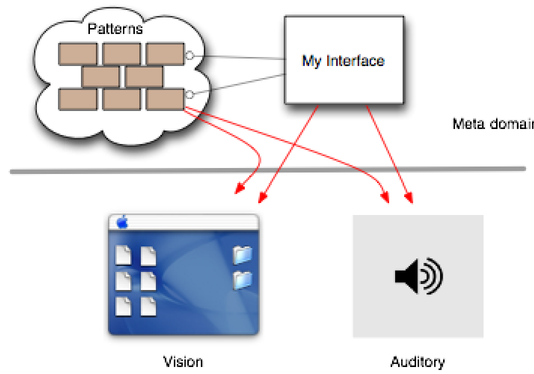



Figure 1: *Mode independent meta domain as common basis*

The approach follows concepts of model-based user interface design (see [22]), but applies design patterns to the process of reification of abstract user interfaces into concrete user interfaces. This was chosen for several reasons: First, the discipline of auditory design is a long way currently from providing formalised standards that could be used to automatically generate code. It lacks libraries or toolkits at the conceptual level that could be used to develop auditory interfaces. Secondly, the specific properties of auditory design makes reification based on rules difficult as the design space lacks heuristics and is aesthetically difficult to manage. Patterns are flexible: *"Each pattern describes a problem that occurs over and over again in our environment, an then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same twice"* [15].

Design patterns also address problems identified in model-based interface design: Due to the strict formalised process, final results are often unpredictable and the high threshold of learning required to use the tools prevent them from being used more widely (see [23]). Because design patterns are instantiated manually, the result can be manipulated directly.

In the following sections we will elaborate the design and implementation of patterns that are concerned with a widely used part of user interfaces: Menus. This part of the user interface was chosen because of flaws we identified in a menu implementation in a previously designed prototype. The evaluation of this prior prototype revealed that the mapping of hierarchical structures like menus onto a static spacial grid is not appropriate for the auditory domain. So finding a more flexible and dynamic representation was the aim for this subsequent design [14].

## 3. AUDITORY MENU PATTERNS

According to Alexander, design patterns are defined through a textual description. Similar to the categories defined by Alexander, we chose the following properties in order to define our mode-independent patterns: The **Name** should be short, but indicative and followed by a **confidence level** that allow authors to indicate how well this pattern is tested and how confident they are with it. A short **Illustration** showing schematically the context of the problem is followed by a textual description of the interaction **problem** to be solved. To every problem apply **forces** that will result in a trade-off in the solution. Furthermore, the problem can be categorised into one or more interaction **principles**. Subsequently we describe the generic and mode-independent **solution** of the problem followed by the **rationale**. In order to build up not only a set of patterns, but a language of patterns, each pattern is linked to others and therefore has a **context** for its use. Finally, there is guidance how to transform this pattern into different modalities. There are **visual examples** and **auditory examples** that show proven real-world solutions to the given problem.

In order to illustrate this approach we developed patterns to create an auditory menu based on the experience we had with using an auditory menu in a prior prototype [14]. The **Command Area** pattern originates from a set of interaction patterns created by Welie (see [19]) and describes the concept of menus through the definition of the following problem: *The user needs to know where to find the possible commands and how to activate them*. For the mi::patterns we created a new sub-pattern that is called **Hierarchical Navigation**, because we realised that this is a common problem not only appearing in menu structures. Table 3 shows the full description of the **Hierarchical Navigation** pattern.

For realising an auditory menu, we use the **Command Area** pattern that subsequently utilises the **Hierarchical Navigation** pattern to solve the problem. According to the guidance given in the pattern, we implemented this pattern as an auditory display as shown in the next section.

## 4. DESIGN & IMPLEMENTATION

The task is to design a menu for real-world applications like a mail client, internet browser, editor or a file manager. Our design is based on the prior prototype in which an auditory menu was realised as part of a auditory file manager application [14]. The components of this application were laid out in a 3D virtual audio environment that simulated a simple box-like room. The user was able to navigate the room using a Joystick. The menu was located on a grid on the left wall with the main entries at the level of the user. When selected they unfolded to the ceiling presenting the contents of the menu which could be selected by ”flying” up to the desired item. The main problem identified in the evaluation of this part of the interface was that users got confused by the rigid grid layout for structured information.

The refined design of the patterns takes these findings into account and replaces the grid layout with a more flexible metaphor. The user is positioned in the middle of a virtual room with a big, horizontal dial in front of him/her. The room is 5 meters per 8 meters and 5 meters in height while the radius of the dial is 7 meters and its centre is located slightly outside the room. The menu items are located on the edge of the dial and the user is able to turn the dial in either direction. In this way users can bring whatever they are interested in to the front of them and select the item. If the item is a submenu, a preview of its contents is given by showing the first two entries at more distant locations. All items are synthesised speech, speaking the item’s title repetitively, a female voice for the main dial and a male voice for the preview items. Figure 2 shows the design. Once the user has brought the desired item to

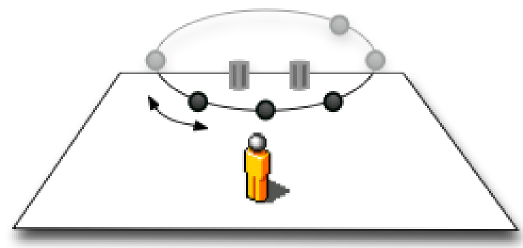


Figure 2: *VR menu implementation with the vertical dial, the menu items and the two preview sources*

the front, it could be selected. If it was an menu entry the system confirms the selection, if it was a submenu, the new menu items are positioned on the dial as before and an ambient whispering voice would silently speak out the root entry the user selected.

The design was supported by various non-speech sounds in order to strengthen the metaphor. The dial makes a rolling sound when moved, entering and leaving sub-menus produces a sound like a Starship Enterprise door. Confirmations for selections are conveyed by a single bell sound. For interaction we chose the left dial on a commonly used gamepad. The reason for not using just the arrow keys on the keyboard was to make sure people distinguish the representation model clearly from the screenreader which uses the arrow keys to navigate.

The prototype was implemented in SuperCollider3[1] and made extensive use of the virtual audio environment extension AmbIEM[2]. AmbIEM allows for rendering multiple sound sources in virtual environments using the Ambisonics algorithm up to 3rd order. It also includes headtracking, but that was not used with this prototype.

## 5. EVALUATION

This evaluation compared the designed prototype against the commonly used screenreader technology. We were interested whether the improved design solved the problems with disorientation in the previous prototype and whether we can actually take significant advantage of the virtual environment and the possibility of presenting multiple menu items at the same time.

### 5.1. Evaluation Test Design

Menus for four well known application types were implemented: a mail client, an internet browser, a text editor and a file manager application. Two of them were used the new prototype, two a

[1] `http://supercollider.sourceforge.net/`

[2] `http://sonenvir.at/downloads/sc3/ambiem/`

graphical user interface based on Apple's Cocoa framework which seamlessly integrates into the VoiceOver[3] accessibility interface of Apple's Mac OS X.

All menus mimic subsets of the corresponding applications on Apple Mac OS X. Their size and complexity have been equalised so that each menu consists of 3 levels and an average distribution of 7, 38 and 10.25 items per level. Every item in the first level was a sub-menu as usual with menus and 3 of the items at the second level would lead to another sub-menu.

Firstly, participants were introduced to the context of the test and were given an explanation of the model behind the prototype. They were asked to answer some questions in a pre-questionaire before they got a 5 minutes training session with the prototype. They were also instructed in the use of the VoiceOver screenreader which only required standard interaction through the arrow keys. Subsequently, the four menus were presented, alternating the prototype and the screenreader. The type of the menu to start with was also varied. Each menu was presented for 2 minutes in which the users were asked to freely explore the menu and report on the type of the application and the menu contents. Before proceeding to the next menu, they were asked to find 5 menu items as quickly as possible. Three of these items are on level 2, two of them on level 3 (in a sub-sub-menu). All actions in the menus were recorded in log files and participants were encouraged to speak out loud during the test which was also recorded on tape. Finally, participants were interviewed in a post-questionaire. The evaluation test took around an hour for each of the eight participants to accomplish. They were all postgraduate students at the college except for one person who is a professional and also the only one with a visual impairment.

### 5.2. Analysis and Results

All participants had long experience with computers and good knowledge of the functionality of all applications used. All of them use computers on a daily basis in their professional lives. Six mainly use Windows, two mainly Mac OS X as operating system, but with the exception of one all had worked with the other and also know Linux. There were 3 females and 5 males taking part, mostly being between 25 and 35 years old with the exception of the visually impaired participant who is in his 40s. They all reported to think of menus as a pull-down structure with multiple levels, one also mentioned a more abstract tree structure. Asked for the most important features of menus, fast access, completeness and good overview were reported most often.

During the training session with the prototype the participants picked up the metaphor of the dial in the virtual room quickly. They got used to the interaction device and on average triggered one interaction every 2 seconds. When exploring the presented menus freely for 2 minutes, participants were already significantly faster with the prototype triggering 1.3 interactions per seconds on average. All participants had a clear understanding about the structure of the menu and the kind of application it represents after this 2 minutes of exploration, although not all found the sub-sub-menus during this phase.

When it came to finding certain items in the menu we saw participants were stable at the interaction per seconds rate. However, it was remarkable that participants needed more time to accomplish the tasks on their second try with the prototype while they further improved with the screenreader. Figure 3 shows these trends. Due

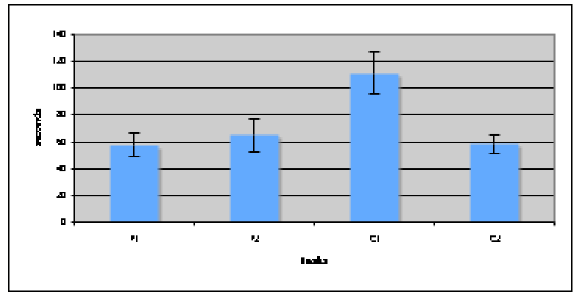


Figure 3: *Time per task (P1/2 for prototype, C1/2 for screenreader)*

to the prior training session, people performed well on the first set of tasks with the prototype, but the concept of the screenreader was not very familiar to most participants and so their performance was poor in the beginning. In the second run, however, participants slightly slowed down using the prototype while the performance with the screenreader approached the level of the prototype in the first run. These numbers support the feedback we got during the debriefing where participants were pointing out that the repetitive speech used for the menu items in the prototype was very exhaustive. People became tired by the redundant information and suggested that a mechanism with which they could trigger the speech would be better. However, that would mean finding a different way of making the sound source location apparent to the user at times when there is no speech output.

There is one notable exception to this trend. The visually impaired participant was the only one to further improve with both systems on the second run. Remarkably, although this user was very trained with the screenreader, the performance with the prototype was not far behind. Figure 4 shows the user's timings.

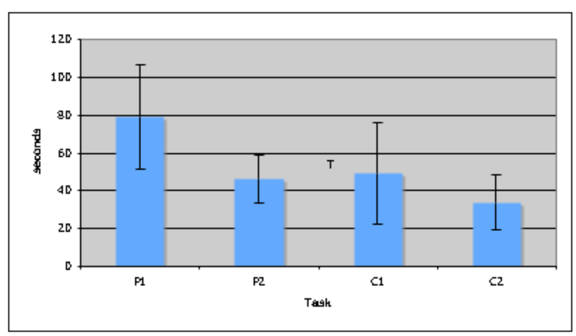



Figure 4: *Time per task (P1/2 for prototype, C1/2 for screenreader)*

In the debriefing questionnaire most participants identified the naturalness of the interface as the major benefit with the prototype in comparison to the screenreader. This was due to the virtual environment and the non-speech sounds that also supported the metaphor significantly. The mental model of the interface was easily adapted to the dial metaphor and lead to a more abstract view of

[3] `http://www.apple.com/accessibility/voiceover/`

menu structures for most users and they never confused different branches of the tree structure.

Participants found the quality of the speech synthesis acceptable, both systems used the built in speech feature of Apple OS X, and despite of one were reporting that the non-speech sound was an important and helpful addition to the interface. The quality of the sound rendering was good and all participants reported that they perceived the sound sources in a rather big room outside their head.

Interaction with the interface was easy in both cases. Participants liked the gamepad which gave it a more game-like character for some of the users. There was, however, one issue that most participants reported: Because the mapping on the gamepad was *forward/up* for *selection* and *backwards/down* to *leave* a submenu, users were sometimes confused because of their mental model of a pull-down menu. This was especially true when they came back to the prototype after using the screenreader where the arrow keys on the keyboard were mapped as expected for a pull-down menu. However, users were able to recover quickly and did not find it a problem.

When people were asked for suggestions for how to improve the prototype 3 people said they had troubles distinguishing leaf items and submenus. The cue to that would have been that there are preview sources active for submenus, whereas there are none for leafs, but users reported they did find that cue too weak. Furthermore, they would have liked to have the menus wrapping to make navigation faster. Two users suggested longer training periods, but the analysis of the data collected showed this would not have helped.

In general people enjoyed the prototype and said it was fun playing around with it. However, the major problem seems to be the repetitive speech that tired users. Although this might not be such a big issue for menus, as they are usually not used over a longer period of time, this is a generic problem of how to tackle the transient manner of sound in spatial environments.

## 6. CONCLUSIONS

In this paper we described our approach to auditory display design through design patterns and described the use of refined versions of patterns we had used in prior experiments. We implemented the auditory display and evaluated it against the existing state-of-the-art audio interface in computers, a screenreader. The improved design removed the flaws we identified in previous work such as the users confusing menu branches and getting lost in the menu structure, but we identified different problems that need to be resolved. We saw a significant degradation of user performance over time that indicated that the repetitive use of speech was very tiring. This leads us to the more general problem of making the user aware of sound sources and their locations in spatial environments without repetition or imposing a lot of cognitive effort on the users. In general, results show that the design is still hardly a significant improvement over existing screen-reader technology.

The current definition of the pattern is based on an existing pattern for graphical user interfaces. It was rewritten for the problem statement and the generic solution to be mode-independent. The proposed auditory solution is the second iteration of an ad-hoc solution that was used in a prior prototype. We can not claim that these patterns represent common practice or valid design knowledge. A more fundamental methodology needs to be developed to obtain this design knowledge to flesh out mi::patterns.

Future research will focus on developing a framework that will allow for capture, validation, application and refinement of design patterns. We aim at filling the gap between existing design knowledge and its application in HCI. We develop methods that allow for obtaining design knowledge from common practice, individual experience and related fields of design and subsequently coding it into design patterns. Valid patterns can then be instantiated and the conveyed design knowledge applied to the design problem at hand. It is hoped that this framework will contribute to building up design knowledge to share within the community or simply allow designers to be aware of and re-use their own designs.

## 7. REFERENCES

[1] Apple, "Apple human interface guidelines," January 2006.

[2] W. W. Gayer, "The sonicfinder: An interface that uses auditory icons," *SIGCHI Bull.*, vol. 21, no. 1, pp. 124, 1989.

[3] A. Walker and S. Brewster, "Spatial audio in small screen device displays," *Personal Technologies*, vol. 4, no. 2, pp. 144–154, 2000.

[4] G. Kramer, Ed., *Auditory Display: Sonification, Audification, and Auditory Interfaces*, Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XVIII. Reading, MA: Addison-Wesley, 1994.

[5] K. V. Nesbitt and S. Barrass, "Evaluation of a multimodal sonification and visualization of depth of market stock data," in *Proceedings of the International Conference for Auditory Display*, R. Nakatsu and H. Kawahara, Eds., 2002, pp. 233–239.

[6] N. Sawhney and C. Schmandt, "Nomadic radio: speech and audio interaction for contextual messaging in nomadic environments," *ACM Trans. Comput.-Hum. Interact.*, vol. 7, no. 3, pp. 353–383, 2000.

[7] B. Arons and E. Mynatt, "The future of speech and audio in the interface: a chi '94 workshop," *SIGCHI Bull.*, vol. 26, no. 4, pp. 44–48, 1994.

[8] G. Kramer, *Auditory Display*, chapter Some organizing principles for representing data with sound, pp. 185–221, Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XVIII. Reading, MA: Addison-Wesley, 1994.

[9] S. Barrass, *Auditory information design*, Ph.D. thesis, The Australian National University, 1998, Adviser-Phil Robertson.

[10] S. Barrass, "Sonification design patterns," in *ICAD Proceedings*, Boston, USA, July 6–9 2003, Internation Conference on Auditory Display.

[11] M. Adcock and S. Barrass, "Cultivating design patterns for auditory displays," in *ICAD Proceedings*, Sydney, Australia, July 6–9 2004, Internation Conference on Auditory Display.

[12] C. Frauenberger, R. Höldrich, and A. de Campo, "A generic, semantically based design approach for spatial auditory computer displays," in *ICAD Proceedings*, Sydney, Australia, July 6–9 2004, Internation Conference on Auditory Display.

[13] J. Lumsden and S. A. Brewster, "Guidelines for audio-enhancement of graphical user interface widgets," in *Proceedings of BCS HCI'2002*, London, UK, 4. - 6. September 2002, British HCI Group.

[14] C. Frauenberger, T. Stockman, V. Putz, and R. Höldrich, "Interaction patterns for auditory user interfaces," in *ICAD Proceedings*, Limerick, Ireland, July 6–9 2005, Internation Conference on Auditory Display, pp. 154–160.

[15] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel, *A Pattern Language: Towns, Buildings, Construction*, Oxford Univeristy Press, 1977.

[16] C. Alexander, *Timeless Way of Building*, Oxford Univeristy Press, 1979.

[17] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable ObjectOriented Software*, Addison-Wesley, Reading, MA, 1994.

[18] J. Tidwell, *Designing Interfaces, Patterns for Effective Interaction Design*, O' Reilly, 1st edition edition, 2005.

[19] M. van Welie, ," 2006.

[20] J. Borchers, *A Pattern Approach to Interaction Design*, John Wiley & Sons Ltd., 2001.

[21] M. Kubovy and D. van Valkenburg, "Auditory and visual objects," *Cognition*, vol. 80, no. 1-2, pp. 97–126, June 2001.

[22] D. Thevenin, J. Coutaz, and G. Calvary, *Multiple User Interfaces*, chapter A Reference Framework for the Development of Plastic User, John Wiley & Sons Ltd., November 2003.

[23] B. Myers, S. E. Hudson, and R. Pausch, "Past, present, and future of user interface software tools," *ACM Trans. Comput.-Hum. Interact.*, vol. 7, no. 1, pp. 3–28, 2000.

**Name** Hierarchical Navigation (2/5)

**Illustration**

**Problem** Users need to navigate through a hierarchical structure to find a desired item.

**Forces**

- do not let users loose focus, being overwhelmed by choice
- convey a mental model for the structure
- minimise the cognitive effort (structure and content)
- maximise the additional information about the items

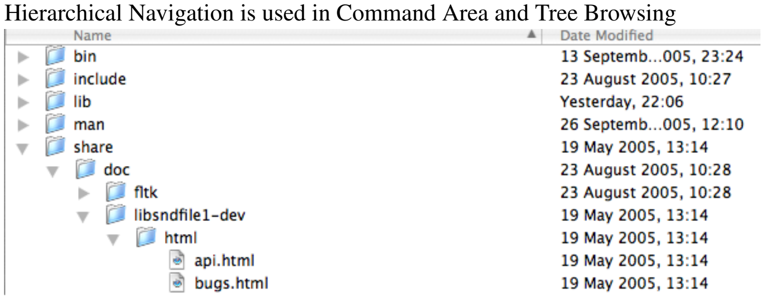
**Principle** grouping, organising information, navigation

**Solution** Make current position and possible choices clear. Hide parts of the structures that the user does not need to be aware of. Employ the most suitable metaphor for the organisation of the information (tree, folders) and provide a shortcut to the origin.

**Rationale** The user may not be aware of depth and complexity of the structure (exploratory use). They need to know where they are in order not to get lost in the hierarchy, but must be able to move quickly through the structure.

**Context** Hierarchical Navigation is used in Command Area and Tree Browsing

**Visual Example**




**Auditory Example** In order to make available choices clear, Hierarchic Navigation needs to provide multiple information sources at once. In a virtual audio environment, the vertical plane is best to maximise the number of items and source segregation. In this example, the choices currently available are spread out on the edge of a big vertical dial in front of the user. This dial can be rotated by the user, making items disappear or coming in from the left or the right. The user may select the item that is in front of her / him. If it is a node containing further items, the new choices are laid out as previously. If it is a leaf, the selection process is accomplished. Items may be represented by any auditory cue (speech, auditory icons, earcons etc.), but should be suitable to form an auditory stream for better segregation. Non-speech sounds are used to distinguish nodes and leafs and to support the mental model of the rotating dial. Ambient sound is used to provide a permanent cue for the origin and/or the current parent-node.

The following recording uses the auditory solution outlined above to present a hierarchical menu structure. All items are synthesised speech, the moving dial sound supports the model and moving through nodes is indicated through a automatic door sound. A selection of a leaf node is indicated by a bell. `http://www.dcs.qmul.ac.uk/~frauenberger/menu-demo.mp3`

**References**

M. van Welie, ”Command Area”, `http://www.welie.com/patterns/gui/command.html`, 2006

A. Savidis, C. Stephanidis, A. Korte, K. Crispien, and K. Fellbaum, A generic direct-manipulation 3d-auditory environment for hierarchical navigation in non-visual interaction, in Assets 96: Proceedings of the second annual ACM conference on Assistive technologies, New York, NY, USA, 1996, pp. 117123, ACM Press.

Table 1: The **Hierarchical Navigation** pattern