# Advanced Auditory Menus

Pavani Yalla & Bruce N. Walker

Georgia Institute of Technology

GVU Center Technical Report # GIT-GVU-07-12

# 1 Introduction

The aim of this long-term research study is to develop guidelines for advanced auditory menus, with an emphasis on mobile devices. The first main focus of the research was to conduct a literature review on the area of menus in general. Studying menus first will give us an understanding of the essential elements and main concepts of menus, which can perhaps then be applied to designing auditory menus. This report is a summary of some of the findings from the literature review. The first part of the report is dedicated to visual menus, while the second part discusses auditory menus in particular.

# 2 Visual Menus

Most technological devices require an exchange of information between the user and the device. Whether it involves the user's navigation through the device's system choices, selection of the device's functions, or input of information, the user needs an efficient way to interact and exchange information with the system. Although command languages offer the user flexibility and power while interacting with the system, they can also strain the user's memory. The user must memorize the numerous commands in order to efficiently communicate with the system. Generally, menus provide an easier solution because our ability of recognition is superior to our ability of recall (Dix et. al, 1998).

## 2.1 General Characteristics

### 2.1.1 Parts of a Menu

Although there are many different types of menus, a few fundamental characteristics of a menu are common to most of them. A typical menu consists of a *title* (also known as a *stem*) and one or more *items*. The title can be a question or a phrase acting as a categorical title. If the title is a question, the items are possible answers to that question. If the title is a categorical designation, the items are options that can be collectively described by the title. Each item could be one of three different types:

*Branch* – Selection of this type of item leads to another menu (a submenu). Generally, the title of the new menu is the same as the branch item that was selected.

*Leaf* – Selection of this type of item does not lead to another menu. Instead, it prompts the execution of some function or procedure.

*Unavailable* – An unavailable item cannot be selected. Its main purpose is to act as a place-holder and convey that the item could become available under other circumstances. It is usually shown faded/grayed out.

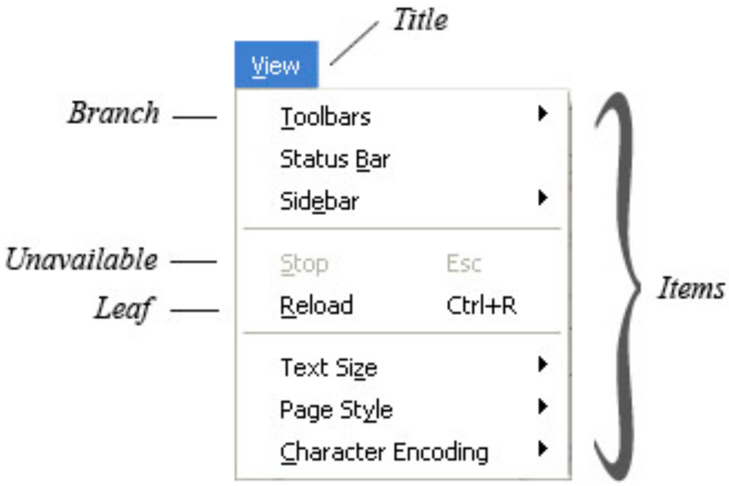


**Figure 1.** An example of a menu frame. The title is "View," followed by eight different items. "Toolbars" is a branch, "Stop is unavailable, and "Reload" is a leaf.

Some additional menu terms are described below and will be used throughout this paper:
*Frame* – A menu frame can be thought of as a single menu which might also be a part of a hierarchical menu structure.
*Node* – Any one of the individual submenus which make up the entire structure of a hierarchical menu.
*Terminal node* – The same as a *leaf*.

### 2.1.2 Types of Menus

The earliest menus were often just lists of options from which the user would have to select one. Menus can no longer be thought of as just lists of options. There are many varieties of menus for various applications. There could be many variations of the same type of menu, and often the same menus are called by different names by different people. Most of the ones described below were mentioned by Norman (1991).

*Single:*

A single menu is the simplest of menu types and is typically just one frame. It presents one set of options and the user simply selects one of the options. Shneiderman (1986) further classifies single menus into binary menus (the simplest of all menus because it only has two items), multiple item menus, and extended menus (single menus which extend onto multiple screens, but still only allow one item to be chosen).

*Sequential Linear:*

A sequential linear menu is a series of single menus which has only one path. The order of the menus within a linear menu is preset, and the user has to make their selections in that preset order. Often, the order has meaning because the individual single menus are related to each other. An example of a sequential menu is a questionnaire where there are a series of questions (name, age, occupation, etc.) that must be answered in a particular order.

*Simultaneous:*

A simultaneous menu is similar to a sequential linear menu in that it is comprised of multiple single menus. However, all of the single menus are visible at the same time. Therefore, the user does not need to make selections in a particular order. They can jump around between each single menu. An example of a simultaneous menu is an online order form with multiple questions and checkboxes. Sometimes, there is not enough screen space for the entire simultaneous menu to be visible. In this case multiple screens or a scrolling mechanism must be used.

*Hierarchical:*

This is one of the most common types of menus used in technological devices today. A hierarchical menu is a tree-like structure of multiple single menus. It is used when there are dependent relationships among the single menus. The selection of one single menu determines which menu will be presented next, and a collection of such dependencies results in a hierarchical menu structure. A hierarchical menu has multiple levels and multiple items at each level. The number of levels is called the *depth* of the menu, and the number of items at each level is called the *breadth*. Although the user has some freedom to choose the way that they would like to branch while navigating, the number of possible branches and the order of the branches is usually preset. Some hierarchical menus allow the user to traverse back up a branch to previous nodes (single menus). If the user makes a wrong selection somewhere in the tree of single menus, they must traverse back up to the node where the wrong selection was made, change their selection, and continue down a new branch of nodes. Many parts of this report will assume that we are talking about a hierarchical menu, unless specified otherwise.

*Connected Graph:*

A connected graph menu is similar to a hierarchical menu. However, the user can jump between nodes and is not restricted to the preset structure of the hierarchical menu. The number of short-cuts between nodes may vary, but in general each node is either directly or indirectly connected to every other node in the structure. Therefore, technically, a hierarchical menu that allows users to traverse back up to previous nodes is a type of a connected graph menu. In essence, a connected graph menu allows shortcuts between nodes where appropriate. Brown (1982) suggests that "if a demand for highly traveled shortcuts arises (measure it!) install direct routes (static GOTOS) where they are needed."

*Event Trapping Menus:*

These menus are a combination of simultaneous and connected graph (or hierarchical) menus. Typically there is a menu bar which presents the titles of multiple menus simultaneously. The user can then chose to mouse-over or click (select) a particular menu title, which then displays that specific menu in the foreground. At any time, the user can easily jump back to one of the other menu titles in the menu bar. This type of menu also provides useful contextual information (the menu bar) in the background while the user is focusing on a specific menu in the foreground. Providing contextual information can be very important when designing a good menu and will be discussed at length in a later section.

*Pie:*

Although pie menus are not very widespread, they offer an interesting alternative to all of the other menus mentioned above. The menu items are arranged in a circular fashion with equal radial distances from the center (which usually displays the title). The cursor's starting position is at the center of the circle, so it only needs to be moved a short distance (equal for all items) to make a selection. In essence, each item in the menu forms a wedge, for which the target size is larger than in a regular linear menu. Therefore, theoretically according to Fitts' Law, the pie menu should result in shorter selection times than linear menus. A research study (Callahan et. al. 1988) found that pie menus do result in shorter selection times, but this study was limited to a fixed menu length of 8 items. The study also acknowledged some of the disadvantages of pie menus: increased screen real-estate usage and a limit to the number of items that can be placed in a circular fashion before the size of the circle becomes impractical. Perhaps due to these reasons, pie menus are not very common in today's technological devices.

*Other:*

Pop-up or Pull-down menus can also fall into several of the above categories. These menus simply appear on the screen as a response to the user's input, usually a click or mouse-over (Shneiderman 1986). Multiple Selection Menus can also fall into several of the above categories of menus. Specifically, they allow the user to select more than one alternative within the same menu (Shneiderman). A fish-eye menu is a variation of a single menu, which uses the fish-eye visualization. As the user hovers over particular items in the menu, that item and the ones surrounding it are magnified. All of the other items are generally very small and practically illegible. This type of menu is particularly helpful with long single menus that take up too much screen space. In a pilot study by Bederson (2000), users preferred fish-eye menus for browsing tasks, but regular hierarchical menus for goal-directed tasks.

## 2.2 Menu Movement Rules

All of the menus mentioned above require the user to bring the item that they would like to select into focus before actually making the selection. Typically an input, such as a key press or mouse movement, is used to move the focus from one object to another. In this section we derive syntax to describe all of the single-step movements which are possible. Each of the menu types described in the previous section can use the same syntax to describe how movement occurs in that particular menu. The syntax is expressed as a series of questions that the designer must consider while creating a menu.

### 2.2.1 Intra-frame Movements

*Intra*-frame movements are all of the possible single-step movements within a single menu frame (excluding shortcuts):

**Table 1.** Possible single-step intra-frame movements where T = title, I = item, n = total number of items, I = index of the item

| | |
|---|---|
| $T \rightarrow I$ | title to item |
| $I \rightarrow T$ | item back to title |
| $I_{i \neq n-1} \rightarrow I_{i+1}$ | item (besides the last item) to next consecutive item, allows scrolling down |
| $I_{i \neq 0} \rightarrow I_{i-1}$ | item (besides the first item) to previous item, allows scrolling up |
| $I_{n-1} \rightarrow I_0$ | last item back to first item, allows a wrap |
| $I_0 \rightarrow I_{n-1}$ | first item to last item, allows a wrap |

For any given menu, the intra-frame movements could be any subset of these movements. When designing a menu, the designer should ask whether the following movements should be allowed:

1 ? $T \rightarrow I$ (title to item?)
2 ? $T \rightarrow I_i$ (title to *any* item? If not, which item(s)?)
3 ? $I \rightarrow T$ (item back to title?)
4 ? $I_i \rightarrow T$ (*any* item back to title? If not, which items(s)?)
5 ? $I \rightarrow I$ (item to item?)
6 ? $I_i \rightarrow I_j$ (*any* item to *any* item? If not, which items(s)?)
7 ? $I_{i \neq n-1} \rightarrow I_{i+1}$ (any item, which is not the last item, to the next item?
8 ? $I_{n-1} \rightarrow I_0$ (last item to first item? -wrapping)
9 ? $I_{i \neq 0} \rightarrow I_{i-1}$ (any item, which is not the first item, to the previous item?
10 ? $I_0 \rightarrow I_{n-1}$ (first item to last item? -wrapping)

The answers to the questions above determine the intra-frame behavior of the menu. Let us take the example of a phonebook in a mobile phone. The title of the phonebook menu might be "Contacts," and the items are all the names entered in the phonebook. If the answer to question 6 is yes, it would mean that the user could jump from any name in the phonebook to any other name with a single key press. If the answer is no, the item to item movement must be defined more specifically. A typical phonebook allows scrolling up and down the items. Therefore, questions 7 and 9 would be "yes." Questions 8 and 10 would also be "yes" if it also allows bi-directional wrapping (the focus moves back to the first item when the user tries to scroll down from the last item and it moves to the last item when the user tries to scroll up from the first item).

### 2.2.2 Inter-frame Movements

*Inter*-frame movements happen between menu frames. There are two different types of inter-frame movements: vertical and horizontal. Vertical inter-frame movements occur between a menu and one of its own submenus. More specifically, this happens when a branch of one menu is selected, and the focus moves to the corresponding submenu. It can also happen if the user moves from the submenu back up to the previous menu. The possible vertical inter-frame movements between a root menu A and a submenu B are as follows:

**Table 2.** Possible vertical inter-frame movements where A and B denote a root menu and a submenu, respectively.

| | |
|---|---|
| $I_{Ai} \rightarrow T_B$ | branch item to submenu title |
| $T_B \rightarrow I_{Ai}$ | submenu title to branch item |
| $T_B \rightarrow T_A$ | submenu title to root menu title |
| $T_B \rightarrow I_{Ao}$ | submenu title to root menu first item |
| $I_{Ai} \rightarrow I_{Bo}$ | branch item to submenu first item |
| $I_B \rightarrow I_{Ai}$ | submenu item to branch item |
| $I_B \rightarrow T_A$ | submenu item to root menu title |
| $I_B \rightarrow I_{Ao}$ | submenu item to root menu first item |

When descending vertically, the focus can move from the branch item to the corresponding submenu's title or to the first item. When ascending vertically, there are even more options. The focus can move from the submenu's title/item to the root menu's title or first item. It can also move back to the original branch item.

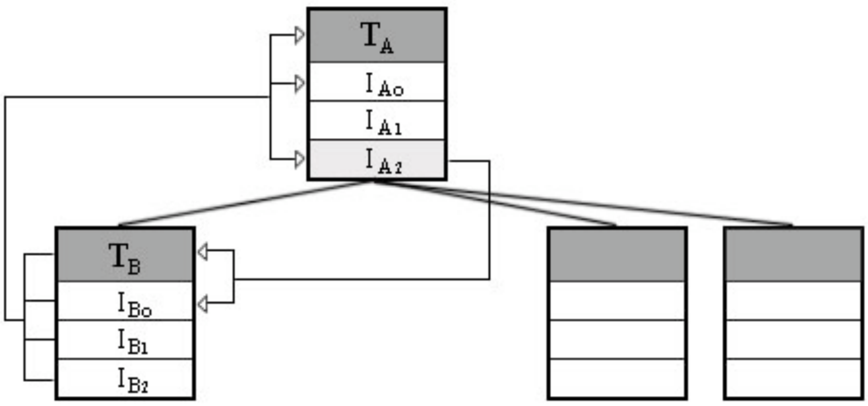


**Figure 2.** Vertical inter-frame movements between root menu A and submenu B.

Horizontal inter-frame movements occur between adjacent submenus. The possible adjacent inter-frame movements between two submenus A and B are as follows:

**Table 3.** Possible horizontal inter-frame movements where A and B denote two adjacent submenus

| | |
|---|---|
| $T_A \rightarrow T_B$ | title of menu A to title of menu B |
| $T_B \rightarrow T_A$ | title of menu B back to title of menu A |
| $I_{Ai} \rightarrow T_B$ | item of menu A to title of menu B |
| $I_{Bi} \rightarrow T_A$ | item of menu B to title of menu A |
| $I_{Ai} \rightarrow I_{Bo}$ | item of menu A to first item of menu B |
| $I_{Bi} \rightarrow I_{Ao}$ | item of menu B to first item of menu A |

The term adjacent implies that the two submenus are next to each other. Technically, this means that their corresponding branch items in the root menu were next to each other. If the designer wants to make the horizontal inter-frame movements wrap, the first and last submenus at that horizontal level should also be considered adjacent.

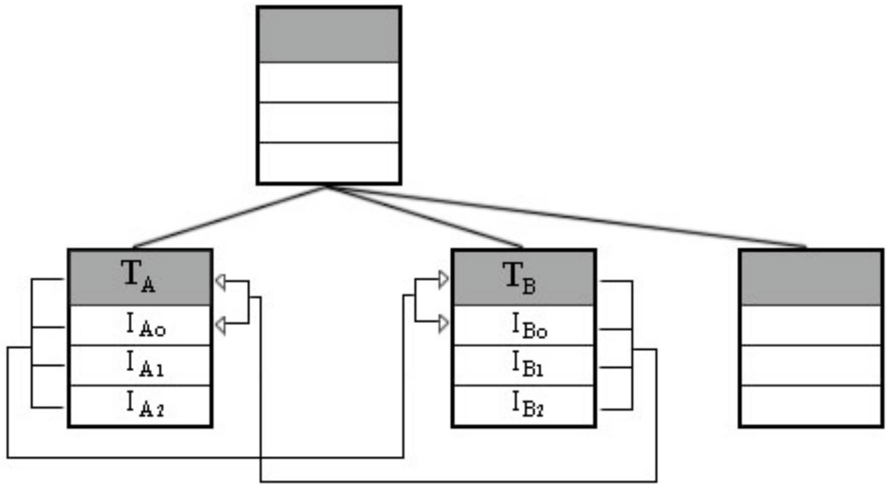


**Figure 3.** Horizontal inter-frame movements between adjacent submenus A and B.

### 2.2.3 Shortcuts

Shortcuts can allow single-step movements to and from anywhere in the hierarchy. They are usually invoked by a unique input and, therefore, might require the use of a new input mechanism. Some shortcuts might use a unique series of existing inputs (key presses or mouse movements, for example), so no additional hardware is necessary.

It is up to the designer to determine where shortcuts are necessary and/or useful. If a certain series of movements within a hierarchical menu are predicted to be used frequently, it might be beneficial to devise a shortcut to that terminal node. Also, the shortcut might be available all the time, or it might only become active at a particular node. For instance, most mobile phones have a shortcut (usually a key press) to put the phone into silence mode. Generally, this shortcut is always available. In contrast, a shortcut which is often found in the phonebook of the mobile phone is the ability to press a letter to move the focus to the first contact name that starts with that letter. Although the names (items of the phonebook menu) might not be next to each other, a jump occurs. Usually, this type of shortcut becomes available only when the user is inside the phonebook.

## 2.3 Cognitive Process of Menu Selection

There are many tasks and subtasks in the menu selection process. Even within a single menu frame, the user must "read the alternatives, choose the desired option, effect the choice, and finally ascertain the consequences" (Norman 1991). These tasks must be evaluated at the cognitive level because they strongly depend on what the user is thinking when they perform the tasks. When studying the cognitive process of menu selection, the timings of individual tasks and the number of errors during these tasks become important measuring tools. Many researchers base their guidelines for menu design on cognitive modeling results. Below, the possible outputs of menu selection are briefly discussed before the various components of menu selection are described:

### 2.3.1 Outputs of Menu Selection

After the user has made the necessary menu movements to arrive at their target selection, a selection has to be made using some type of input device (discussed later). The selection can result in one of two different types of outputs: branching outputs or procedure outputs. Branching outputs cause a new submenu to open and occur if the selected item was a branch item. Often the sole purpose of a menu item is to be a branch and define the navigational characteristics of a menu. If the selection leads to a branching output, the menu's vertical inter-frame movement rules determine where the focus will land in the new submenu (title or item).

If the selected item is a leaf, the output is a procedure output. A procedure output means that some procedure is executed. The procedure could be many different things: display information, input information, perform an internal function, etc. There might or might not be a change in the display while the procedure is being executed. For instance, a dialog box might open up and prompt the user to input information which the system would process to perhaps perform another function. Or a simple display window might open up to display requested information. An internal function such as "save" might not visually change the display of the screen while it is being performed, but some type of feedback should be given to the user to indicate that the procedure has been executed. The concept of feedback will be discussed in more detail in a subsequent section.

Once the procedure has been executed and some feedback has been given, the menu must return to a state in which some title or item is under focus. This is called the "point of re-entry" after a selection has been made. With branching outputs the vertical inter-frame movement rules determine the point of re-entry. With procedure outputs the point of re-entry could be one of several options. It could return to the last item under focus (the selected branching item), or it could jump to another location in the menu. In this way, a procedure output could also be a branching output because it could lead to a different submenu.

### 2.3.2 Components of Menu Selection

*Visual Search*

Before making a selection, the user must visually search the available alternatives for the intended target (depending on whether it is a hierarchical menu, the target of a single menu frame might not be the final target, but just an intermediate one). Users might have their own personal strategies for conducting the visual search. However, researchers have attempted to theorize these strategies.

Norman (1991) seems to support a serial processing theory. When the user already has a clear target in mind, the search process is slightly different from when the user only has a partial idea of the target. With a clear target in mind, the user mentally matches each alternative with the target until a match is found. With a partially specified target, the user must evaluate each alternative and decide whether or not it is a potential match. In this case the user might stop searching once a "close enough" match is found, or they might search the entire list before determining which alternative is the best match. The second option takes more time, but is probably more accurate (Norman). Therefore, the first method is used when speed is important to the user, while the second method is used when accuracy is more important (Norman). In addition, eye-tracking studies (specifically on a pull-down menu) have shown that visual search tends to occur top-to-bottom and is rarely random (Byrne, Anderson, Douglass, & Matessa 1999).

Hornof and Kieras (1997) disagree with Norman's serial processing theory. Instead, they propose that people use both random and systematic approaches for searching, and they process multiple menu items at the same time. In addition, Hornof and Kieras (1999) used cognitive modeling to show that users often anticipate the location of menu items even before the menu appears. Specifically, they can anticipate the position of items higher up in a pull-down menu more accurately than those which are lower.

Regardless of which search strategy is used, in general, repetitive exposure to a menu makes the searching process for that menu faster. Furthermore, alternatives which are too similar to each other cause more processing time and errors. Therefore, phrasing of alternatives should be done very carefully (see Designing Menus section). Ordering of menu items is another very influential factor when it comes to searching the items. This is also discussed in the Designing Menus section.

*Choice Response*

According to Norman's description of searching, the choice process can occur either after the entire set of alternatives is scanned, or during the scanning process. Once the user has found a match to their target and has decided that it must be selected, a response must be given so that the device can detect the choice. The response can be given in various forms, depending on the type of menu and the hardware input components available. Pointing and selection by code are two different selection methods. Pointing can be accomplished with hardware such as a touch screen, mouse, cursor keys (e.g. keyboard, mobile phone), joystick, etc. Selection by code is usually only done with the use of pressing keys. Selection by code is more complicated than pointing because it requires encoding and then decoding, whereas pointing has been practiced since infancy (Norman 1991). According to Norman, among the different pointing devices, touch screen is the most intuitive, followed by mouse and joystick, and finally cursor keys.

Pointing devices which use continuous movements (mouse or touch screen) rely heavily on Fitt's Law to define the time that it takes to move to a target (a logarithmic function of the ratio of the distance to the target and the target's width). Cursor keys use discrete movements, and are governed by the x and y distance between the cursor position and the target. However, other factors also influence the time to select a target. The physical mapping of the input hardware to the menu items is an important factor. For example, a mouse is generally moved on a horizontal plane, while the corresponding menu might be on the vertical plane of a computer screen (this is one reason why touch screens are more intuitive than using a mouse). Perlman and Sherwin (1988) suggest that menu display formats should match input device formats, and give the example of a horizontal key layout corresponding to a horizontally laid out menu. Some software applications violate this guideline when they visually arrange their function options in vertical columns, but the keyboard has the function keys arranged in a horizontal row.

Many experiments have been conducted to determine the fastest and most error-proof selection method using a mouse. Kobayashi and Igarashi (2003) propose two methods for increasing the usability of cascading menus. First, they suggest that cursor movement should direct menu behavior (horizontal movement to open or close sub-menus, vertical movement to change the position of tentative selections). Second, they suggest that the sub-menu open up at the position where the horizontal movement occurs. Most current cascading menus require the user to hover over a small arrow next to a menu item to open a sub-menu (which is opened adjacent to the main menu to avoid overlapping). Kobayashi and Igarashi found that their suggestions reduce selection times and shorten search path lengths when compared to current cascading menus.

*Response Feedback*

The device should give feedback to the user once a selection has been made. The feedback can be textual, graphical, or auditory (discussed later in the paper). Sometimes, the result of the selection is immediately apparent visually. This is the case when the selection output is a branching output. However, if the function of the selection is to perform a procedure, the success of the selection might not be immediately apparent to the user. Therefore, the system should display a message that indicates the status of the procedure which is being executed ("Calling John..."). Prompt feedback can reduce menu navigation times in hierarchical menus because the user can quickly move ahead or go back depending on the negative/positive feedback provided by the system. Visual feedback is discussed further in the Designing Menus section.

## 2.4 Designing Menus

A menu structure might consist of several menu screens, depending on the type of menu and the amount of information. The menu's visual layout, content organization, and graphic design elements are all important aspects that must be considered during the design. Since users might not read the entire menu word for word, the design of the menu should help them to accomplish their goals whether they are simply browsing or searching for a specific item.

### 2.4.1 Content Organization

There are often trade-offs that have to be made while determining how to organize the content in a menu. One of the most basic issues involves the amount of information that should be present on the screen at one time. Too much text or information on the screen at the same time hinders the user's ability to scan and process the information. A cluttered screen will require the user to take more time to find the desired target. However, less information on a screen might mean that the information has to be divided into several screens. Additional screens imply additional steps necessary to reach the user's goal. This also may have a cognitive effect on the user's performance. This trade-off between the amount of information on a screen and the number of screens is especially applicable to a hierarchical menu. It then becomes a trade-off between depth and breadth, which deserves some discussion.

*Depth vs. Breadth in Hierarchical Menus*

At some point the designer of a hierarchical menu has to decide whether to include many alternatives at a single level or to split the alternatives up among more levels. The number of levels in a hierarchical menu is referred to as the depth. The number of alternatives at any one level is referred to as the breadth. As the breadth is increased, the user has more alternatives to choose from, and the time taken for the cognitive matching that takes place between each alternative and the user's target increases. However, if the designer chooses to divide the alternatives into smaller groups among more levels, the increased depth also poses an issue. More levels mean that the user must make more selections from more frames before reaching their target, which also increases the time.

Numerous studies have been done to determine whether broader or deeper menus are more efficient, and different researchers have found various results. One study found that error rates increased from 4% to 34% as the depth of the menu increased from one level to six levels (Snowberry et. al. 1983). Miller (1981) found that eight item menus with a depth of two levels were better (less errors and faster target selection) than a two item menu with a depth of six levels, a four item menu with a depth of three levels, or a sixty-four item menu with a single level. Other researchers found similar results, supporting broader and less deep menus (Dray et. al 1981; Kiger 1984; Shneiderman 1986). However, a recent study appropriately states that the menus structure should be adapted to the screen size and complexity of the task (Chae & Kim 2004). For instance, menus on a cellular phone should definitely have different depth/breadth guidelines than menus on a desktop computer. According to Geven, Sefelin, and Tscheligi (2006), users prefer a deeper hierarchy to a broader one on mobile phones.

*Order of Menu Items*

The ordering of menu items within a menu is also very important. If the ordering follows some logical scheme, it can facilitate the user while searching the various alternatives for the target selection. Depending on the purpose of the menu and the information contained in it, a particular ordering scheme might be more appropriate than another. Alphabetical ordering might be used when the items are keywords or phrases that the user might already have in mind, therefore, making it easier for them to find. In a study conducted by Card (1982), alphabetical ordering was found to be superior to random ordering. Numeric ordering is appropriate when the alternatives are numbers such as size. Chronological ordering is appropriate when the alternatives have to do with time or dates. Sequential ordering can be used when the items follow a cognitive order or order in a particular process.

Items can also be ordered based on the frequency of use. More frequently selected items would be placed higher than less frequently selected items. This is done as an attempt to decrease the time it takes to find a target, especially if searching of the items is done serially. The frequency of use might be determined ahead of time by the designer (based on estimates and predictions), or it might be determined dynamically while the system is being used. In the latter case, the system would have to learn the selection patterns of the users and reorganize the items based on the frequency of selection. Menus in which more frequently used items are placed at the top of the menu are sometimes called "split" menus. Sears and Shneiderman (1994) found that performance times were reduced by 17-58% (depending on the site and menus) when split menus were used as opposed to alphabetical menus. However, it had also been found that split menus which had been generated dynamically could result in poorer performance, perhaps because the menu might look different each time (Mitchell & Shneiderman 1989). Therefore, lack of consistency might undo any benefit provided by sorting the items based on frequency.

### 2.4.2 Providing Action Feedback

When a particular action takes place, whether it is a change in state, a movement, or a selection, the user should be given feedback. Providing feedback is a fundamental interface design rule. For menus this can be accomplished through the use of visual, audio, or haptic feedback. This section will focus on visual feedback.

Visual elements of the menu design often facilitate the user (both consciously and subconsciously) while navigating and making selections in a menu structure. Visual characteristics such as size, color, transparency, highlighting, and outlining can really enhance both the aesthetics and functionality of the menu.

For example, when a menu frame is activated, it is important that the graphics capture the user’s attention. The view of the new menu could be one of several different types. It could take up the entire screen or pop-up/down over the existing screen. Taking up the entire screen would immediately and effectively capture the attention of the user. However, there would be a loss of contextual information because no information from the previous state would be visible. A pop-up/down menu can capture attention in varying levels. Although the menu is superimposed on top of the existing screen, the existing screen (background) can be grayed out to indicate that the pop-up/down menu is the active state. This can capture and retain the user's attention fairly well. If the background is not grayed out, but at least distinct from the current menu itself (different font, outline around current menu, change in color, etc.), this also helps the user focus their attention on the current menu frame.

Feedback indicating the focus within a menu is even more important. This feedback aids the user with movements made before a selection is made. Highlighting or outlining of the focused title or item is often used. When the focus is on a desired item, the user might want to select it. Feedback should also be given to indicate that a selection is being made. The highlight/outline might change color, get darker, or flash on and off while the item is being selected. If an outline was used to indicate focus, perhaps a highlight could indicate selection.

In essence, there should be some visual distinction between the feedback given for a focused item and a selected item.

Occasionally, text is used to provide action feedback. For example, some menus display the text of the focused item in another section of the screen. As the user moves the focus, the displayed text also changes and draws the user's attention. Another example is when a textual message is displayed to emphasize a selection and/or its consequences.

### 2.4.3 Providing Contextual Information

Contextual information can be very important to the user when the menu is complex. For example, when a hierarchical menu has many levels, it is natural for the user to become disoriented and not know their exact location in the hierarchy. Location information can be important especially when the user wants to retrace their steps back to a previous level in order to branch to a different path. Good contextual information can be provided by carefully designing the way menus are displayed.

Within a single menu frame, the size of that frame is often very apparent to the user, especially if the number of menu items is small. Some menus have too many items to fit on one screen, and multiple screens are necessary. In this case the user is usually still given visual information to convey the number of items and the location of the focus. For example, most address books in mobile phones display a scrollbar which gives the user an idea of how far along they are in the list. Although the user cannot move the thumb of the scrollbar to control the screen display, it is still useful for the contextual information that it provides. The size of the thumb sometimes conveys the length of the menu (a smaller thumb implies a longer list and a larger thumb implies a shorter list).

Within the entire structure of a hierarchical menu, each menu frame should also provide information about the user's past selections which led them to the current frame. This will assure the user that they are progressing toward their target, in case they forget what selections they already made or even what their target is. During vertical inter-frame movements, for example, instead of replacing the entire screen, successive menu frames can overlap previous frames. Since the previous frame are still visible, this gives the user information about how many levels they have gone through and, depending on how much of the previous screens is visible, what previous selections were made. During horizontal inter-frame movements, displaying at least the titles of each submenu at that horizontal level can provide valuable information. Sometimes tabs are used for this purpose. Even using a different background color for the submenus at each level can help the user subconsciously realize which level they are at in the hierarchy.

Another good design practice is providing as much information as possible about the outcome of a selection (before the selection is made). This adds to the predictability of the interface and can be done in several ways. Simply indicating whether an item is a branch, leaf or unavailable item gives the user an idea of what will happen if it is selected. Some computer menus display an arrow next to branch items to indicate that they lead to another menu. Unavailable items are usually grayed-out to convey that they cannot be selected. Another useful feature that leads to predictability is displaying the submenu of a branch item when the branch item is under focus. Therefore, the user does not need to actually select the item to see the submenu. And since they are still in the root menu, they can still move the focus between all the branch items before selecting the one they think is correct.

Consider the cascading menu in Figure 4. It provides many of the different types of contextual information discussed above: vertical inter-frame movements, horizontal inter-frame movements, type of leaf, and focus-induced display of submenus.

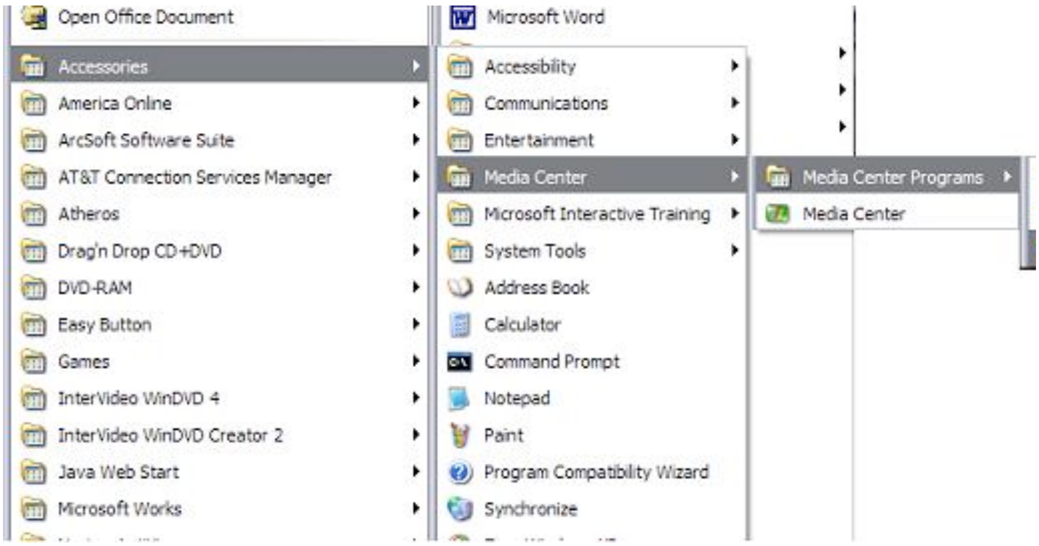


**Figure 4.**: A cascading menu on a computer.

Contextual information can also be conveyed with text by using verbal linkage. For instance, making the user's previous selection the title of the current frame reminds the user of their previous selection. Additionally, a line of text which shows all of the previous selections can be used to keep track of the user's navigation. In the figure below, the title of the current frame, "Sample Pictures," was the last selection that was made. Also, the address bar provides the text for each previous selection that was made.

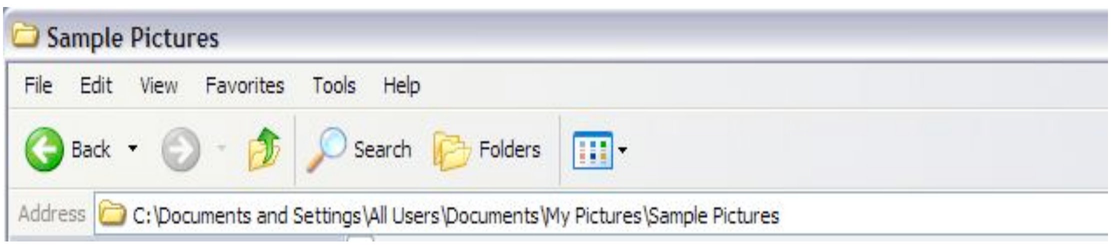


**Figure 5.** Menu that uses verbal linkage to provide contextual information (see title and address bar).

Shortcuts were discussed in a previous section. In order to take advantage of these shortcuts, the user must know that they exist. It is possible to indicate the shortcuts on the input hardware itself (labels on keys). Good menu design, however, also displays the shortcuts available on the display screen. A simple way to do this is to use text, indicating how the shortcut should be invoked, next to the item which would be the result of the shortcut (as shown in the figure below).

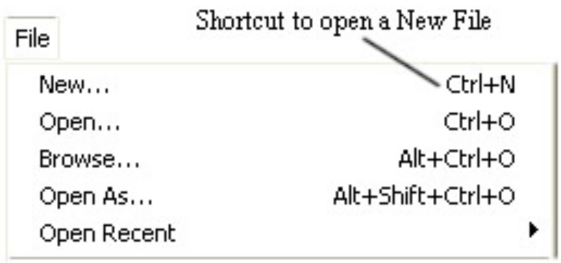


**Figure 6.** The shortcuts are displayed next to their corresponding items.

### 2.4.4 Consistency

When designing a menu, consistency should be practiced wherever possible. Grammatical consistency among titles and items can prevent the user unnecessary confusion. Some guidelines for phrasing items were given by Shneiderman (1986): use consistent and familiar wording, make alternatives distinct from each other, be concise, etc. When designing the menu frame, it is also very important to be consistent in the placement of text between frames. A study by Teitelbaum and Granda (1983) showed that varying the position of information such as titles caused the user's thinking time to almost double.

Consistency is especially important in a hierarchical menu structure. The feedback and contextual information provided should be consistent across all submenus. If a highlight is used to indicate focus, and a flashing highlight is used to indicate selection, this scheme should be used in all the submenus of the entire hierarchy.

In general, the movement rules should be consistent across all submenus (with exceptions when necessary). For instance, let us imagine that the movement rules for a particular hierarchical menu dictate that the point of re-entry after a leaf item has been chosen is the same leaf item. This means that the focus should land back on the leaf item once its corresponding function has been performed. This rule should apply in all such situations throughout the entire hierarchical structure. However, there could be exceptions. Consider the phonebook example again. The user selects the item "Kathy" with the intention of deleting it from the list. When the name is selected, a submenu is opened with "Kathy" as the title and the following options as items: *edit* (branch), *save* (leaf), and *delete* (leaf). The user selects *delete*, and that name is deleted from the phonebook. According to the rules just described, the point of re-entry should be in the "Kathy" submenu, with the focus on *delete*. However, since "Kathy" was just deleted from the entire hierarchy, the submenu associated with "Kathy" was also deleted. Therefore, the point of re-entry goes back to the phonebook's root menu. This is just one exception to the consistency principle, but there could be others. The designer must carefully determine when consistency can be practiced.

# 3 Auditory Menus

Compared to the amount of literature available on visual menus, there is very little available on auditory menus. This is due to the fact that auditory menus are a relatively new concept. Very few researchers have conducted meaningful studies in the area of auditory menus (Gaver and Brewster, to name a couple). Even more noteworthy is the fact that there are very few, if any, comprehensive sets of guidelines for auditory menus. In general, auditory interfaces have the potential of making devices which rely solely on visual displays more usable and accessible to a wide range of users (including users with visual impairments). The United States Federal Government acknowledged this potential, and made it a requirement for visual interfaces to have an alternative interface (Telecommunications Act Accessibility Guidelines of 1998; Electronic and Information Technology Accessibility Standards (Section 508) of 2000). However, the design guidelines suggested are not specific enough for most designers. Therefore, our goal is to develop specific guidelines for auditory menus. We intend to conduct the research with a universal design approach, but specifically with visually impaired users in mind. First, however, it is necessary to discover the research that has already been done in this area. The rest of this report summarizes the past research and, in addition, suggests ideas for conducting new experiments.

## 3.1 Speech in Auditory Menus

When people think of auditory menus for technological devices, they usually think of speaking menus. Advancements in Text-To-Speech (TTS) and voice recognition technologies have been especially helpful to visually impaired users. Specifically, screen readers are software applications that interpret what is being displayed on a computer screen. The screen reader uses a TTS engine to convey the output with synthesized voice. Many screen readers also support Braille output. JAWS from Freedom Scientific and Window Eyes from GW Micro are two popular screen readers. They allow a blind person to use all Windows applications as well as browse the Internet.

The visually impaired have greatly benefited from similar technologies in mobile phones. There are two categories of accessible phones for this population: off-the-shelf phones and phones with the Symbian operating system. Off-the-shelf phones come with a built-in speech output feature. The following are a couple of off-the-shelf accessible phones currently available on the market:

*LG VX 8300, available from Verizon Wireless:* This phone comes with the Voice Command feature which allows users to control many of the phone’s functions with their voices (using voice recognition). They can speak the phrase “call someone” and the phone will respond with a prompt to say a person’s name. The user will then say a name in their address book and the number will dial. The Voice Command feature can be disabled, and the phone will simply speak the name of each item as the user scrolls through the menu options.

*Owasys 22C, from Capital Accessibility:* This phone is a screen-less phone designed specifically for blind or low-vision users. Aside from its speaking capabilities, since it was designed specifically for the visually impaired, the hardware of the phone is quite accessible as well. The keys on the keypad are rounded and spaced far apart from each other. This makes them easily identifiable by touch. Also, there are a few extra keys that make it more accessible. For example, the “Status” key can be pressed to get status information such as the battery level.

Phones with the Symbian operating system have the ability to install third-party software that converts the phone’s display text to speech. This is similar to how screen readers on computers work. Phones with such software generally offer more functionality than off-the-shelf phones. The TALKS software by Nuance is an example of such software. It runs on Symbian Series 60 phones. Since it is practically a screen reader, TALKS speaks everything that is on the screen and can go very deep into menu structures. A visually impaired user can use practically all of the functions available on the phone. Although a high level of functionality is possible, it is not always practiced because many of the functions might still be difficult to execute. This often depends on the user’s level of training and personal needs. More user testing should be done, followed by improvements addressing user suggestions.

Most screen readers have variable speed settings. Therefore, if the user is capable of comprehending the speech at a faster speed, they can change the setting and reduce the amount of time it takes to interact with the system. Variable speed settings are especially useful because some blind users are able to comprehend speech at very high speeds. Asakawa et. al. (2003) found that expert blind users were able to comprehend TTS speeds that were higher than the highest rates of most TTS engines. They suggest that TTS engines should support faster speeds.

## 3.2 Non-Speech Sounds in Auditory Menus

Although speech is the most obvious solution for conveying menu content, non-speech sounds have also been found to be effective in many different ways. Auditory icons, earcons, and spearcons are the specific types of non-speech methods described in this section. Although most of these methods have not been implemented outside of the research environment, the research is ongoing, and there is much potential for future implementation in industry as well.

### 3.2.1 Auditory Icons

The first guidelines for auditory icons were developed by Gaver (1986). Auditory icons are auditory representations of visual icons. The sound for each auditory icon is usually an everyday sound that can be associated with that icon. Gaver's SonicFinder, an auditory interface that he developed at Apple Computer, uses the auditory icon concept (1989). For example, take the example of a person deleting a file from a computer's desktop. Imagine a user dragging the file across the desktop to drop it in the trash can. The corresponding auditory icons could be a scraping sound while the file was being dragged, the sound of shattering dishes when the file is dropped in the trash can, and a crunching sound when the file is actually deleted. The frequency of the scraping sound would vary with the size of the file being dragged. Gaver's auditory icons depend heavily on metaphors in the everyday world. Such metaphors might not exist for every icon on a graphical user interface. Logical sounds do not exist for every single computer command or action, and arbitrary sounds must be used as well. Therefore, the synthesis of auditory icons can be difficult.

### 3.2.2 Earcons

Earcons are described as "non-verbal audio messages that are used in the computer/user interface to provide information to the user about some computer object, operation or interaction" (Blattner, Sumikawa, & Greenberg 1989). The first real tests using earcons were conducted by Brewster, Wright, and Edwards in 1992. They describe earcons as "composed of motives, which are short, rhythmic sequences of pitches with variable intensity, timbre and register" (Brewster et. al. 1992). Their first set of experiments determined that earcons are better than unstructured bursts of sound, and that people can recognize the earcons well if the pitch, rhythm, and timbre are used appropriately. Initial guidelines were also proposed for the use of these sound qualities.

Extensions to these guidelines were put forth in a subsequent paper (Brewster, Wright, & Edwards 1995). The paper derived guidelines for earcons based on the following: timbre, register, pitch, rhythm/duration/tempo, intensity, spatial location, making earcons attention-grabbing and compound earcons. The most important development from another set of experiments conducted in the same year was the fact that users could recognize parallel compound earcons just as easily as serial compound earcons. An example of a serial compound earcon is the sound heard for "open file" when the earcon for "open" is played, followed by the earcon for "file." The same phrase "open file" can be conveyed with a parallel compound earcon which would play the earcons for both "open" and "file" at the same time. Therefore, the parallel compound earcon would be shorter. If parallel earcons are as easily recognized as compound earcons, designers can just use parallel earcons and save time. This study also showed that non-musicians recognized earcons just as well as musicians did.

Earcons were also found to prevent slip-off errors when they were used to sonify graphical buttons (Brewster 1996). According to Brewster a slip-off error can occur when the user perceives that a button selection has been made after the mouse has been clicked on the button. In reality, however, the mouse must also be released for the selection to actually take place. If the user’s mouse accidentally slips off of the button before it is released, the selection has technically not been made, but the user might think that it has. In order to alert the user that a slip-off has occurred, Brewster suggests playing different types of sounds for a correct selection and slip-off error. Similarly, earcons were later found to be helpful in providing feedback about menu and item slips, thus improving usability (Brewster & Crease 1999).

Perhaps the most important development involving earcons (for our purposes) is the discovery that earcons can provide navigational cues in a hierarchical menu (Brewster, Raty, & Kortekangas 1996). In the hierarchical menu structure, each node is an earcon. Each earcon inherits the properties of the nodes above it, but adds an extra sound parameter to indicate the next level. Therefore, it retains information about the location of that node. Results of the experiments showed that participants identified their location in the hierarchy with over 80% accuracy. They could also identify where previously unheard earcons would fit into the hierarchy with 91.5% accuracy (indicating that the participants learned the rules well). Brewster extended this study to include menus in telephone-based interfaces (1997). Due to the lower sound quality of telephone-based interfaces, recall rates reduced significantly. However, when the earcons were redesigned especially for the telephone interface, recall rates improved to 73% recalled correctly. Participants' recall rates were still 73% even after a week passed, so they could use the system infrequently, and still remember the rules for navigation.

The first experiment using hierarchical earcons (1996) was conducted again, but with newly designed compound earcons (Brewster, Capriotti, & Hall 1996). With the earlier version of earcons, the designer would quickly run out of new sound parameters for each new level that was added in the hierarchy. Therefore, serially played compound earcons, which could be concatenated each time a new level is added, were used instead. The resulting recall rate of 97% was even better than the original experiment. Identification of previously unheard earcons was also 97%.

### 3.2.3 Spearcons

Spearcons were first introduced by Walker, Nance, and Lindsay (2006) as an alternative to auditory icons and earcons. Spearcons are produced by speeding up regular speech. This is done by first converting the text of a menu item to speech using text-to-speech technology, and then compressing the speech while maintaining pitch. At the end some hand tweaking of the audio can be done, but this is not necessary. In essence, each spearcon forms an acoustical fingerprint for each textual phrase.

Walker et al have collected evidence to support the fact that spearcons are more efficient than both earcons and auditory icons when used as enhancements in auditory menus. As mentioned earlier, auditory icons do not exist for all textual phrases, and their synthesis can be very difficult. However, spearcons can be easily created on the fly, and it is possible to create a spearcon for any textual phrase. Earcons are much harder to create than spearcons, and they are considered *brittle* to overall changes in menu hierarchy (Walker et. al., 2006). For instance, when a new item has to be added to the menu in the middle of the hierarchy, either the hierarchical order of the earcons must be changed, or the earcons below the new menu item must be relearned. Since spearcons are not dependent on this hierarchical structure, they are more flexible and tolerant of menu changes. In addition, spearcons provide less arbitrary mappings than earcons because they retain some of the characteristics of the original speech. This should make them easier to learn and remember (experiments are being conducted to support this claim). The spearcons also form natural groupings which could help the user. Consider an address book in a mobile phone. The contacts' names are usually organized alphabetically. As the user scrolls down the list of contacts, they would hear the spearcons for each name. They would only need to listen to the beginning of each spearcon to determine which letter they were at. All of the names starting with "A" would start off similarly; all of the names starting with "B" would start off similarly; and so on. If the user wanted to call someone named "Sarah," they would simply scroll continuously until they heard the "S" sound. Once in the "S" region, they could listen to each spearcon completely before making their selection.

In their experiment Walker et. al. compared search time and accuracy of menu navigation using speech only, hierarchical earcons, auditory icons, and spearcons. The results showed that spearcons were faster and more accurate than the other conditions (followed by speech only, auditory icons, and hierarchical earcons, in that order). A subsequent study found that *learning* rates for spearcons were much faster than for earcons (Palladino & Walker 2007). Auditory icons were excluded from this study because they are difficult to synthesize for realistic menus in technological devices (e.g. computers and mobiles phones).

### 3.2.4 Sonification in Mobile Phones

This section discusses the use of non-speech sounds in mobile phones (phones with speech were mentioned earlier). Mobile phones offer a unique platform for auditory menu research. For instance, screen space has always been an issue with menu design on mobile phones. With auditory menus screen space is no longer a restricting factor, so the overall menu structure need not follow prior guidelines. For example, the suggestion that mobile phone menus be deeper and less broad (Geven et. al. 2006) is not necessarily valid for auditory menus on mobile phones.

Brewster, Lepatre, and Crease (1998) suggest that mobile phones can benefit from earcons. They claim that non-speech sounds can improve mobile phone usability without having to increase screen space. Specifically, they mention the use of earcons for navigation through cell phone menus. In 2000, Lepatre and Brewster conducted actual experiments on a mobile phone to support their previous claims. They sonified the menu system in a Nokia 6110 phone and found that menu navigation performance was higher when participants used the sonified version, as opposed to the regular phone. A subsequent study investigated the subjective reactions of participants who used sonified mobile phones for three weeks (Helle, Lepatre, Marila, Laine, 2001). Despite some encouraging feedback, it was found that most users did not prefer the sonified mobile phone because it was too disturbing. The researchers concluded that further research is necessary to determine the best design of sounds used in the sonified mobile phone. Marila (2002) conducted further studies with sonified mobile phones and found that a simpler sound scheme than the one used previously enhances users' performance. However, more studies have to be conducted to determine specific guidelines that would make the sonified mobile phone more efficient

Furthermore, the hardware and software limitations of current mobile phones must be taken into account when designing for audio. Unlike computers and other devices which use menus, mobile phones are more likely to be used in noisy outdoor environments which could interfere with the sonification. This must also be acknowledged when deriving guidelines for the sounds.

## 3.3 Spatial Audio

Cherry (1953) made one of the earliest efforts to demonstrate a person's ability to single out a preferred source of speech while other sources are present ("the cocktail party"). In an experiment where two different samples of spoken messages were presented in each ear of the participants, the participants were able to pay attention to one of the messages and recall it accurately. However, when asked about the rejected message, the participants could recall very little about it. Many could not even recall the language in which it was spoken. However, when asked whether it was a male or female voice, they could often identify it correctly. Therefore, although the participant was paying attention to a single message, there are certain properties of the "rejected" message which are also recognized.

More recent work has been done concerning using spatial separation of sounds to convey separate items in a menu simultaneously. Discrete items in a visual menu can be perceived simultaneously, however discrete items in an auditory menu are often perceived serially (the audio is played serially to avoid interference between items). Experiments conducted by spatially separating auditory items (using stereo panning or 3D audio processing with headphones) showed that these techniques are feasible for multi-item presentation (Lorho, Marila, & Hiipakka 2001). The same authors used this technique to test its feasibility in conveying breadth and depth of hierarchical menu structures (Lorho, Hiipakka, Marila 2002), and found that spatial audio can indeed convey this information. Walker and Brewster (2000) used spatial audio to sonify a visual progress bar in a small screen device and found that the sonified version provided a useful and effective alternative to the visual one.

Aside from using spatial separation to convey menu structure (spatially separating each menu item), it might simply be used to create separate channels of information. For instance, one channel might be playing the text-to-speech of menu items, while another one plays beeps and other non-speech sounds in the background to convey navigational information. If the sounds are designed properly, the results of Cherry's experiments (1953) might have very important implications. Users would primarily listen to the first channel, but might also notice important changes in the second channel.

## 3.4 Auditory Menus and Visual Menus

A visual interface conveys information in a primarily parallel manner. On any single visual screen the user can see several visual objects (title, items, contextual information) simultaneously and continuously. In contrast, an auditory interface inherently conveys information serially (playing successive sounds based on where the focus is at any particular time). Although it is possible to make a visual serial menu as well as an auditory parallel menu, this takes more effort on the part of the designer. There is also a tradeoff associated with conveying information serially versus in parallel. Typically, serial communication takes more time than parallel communication, but parallel communication can lead to clutter or information overload. Therefore, the tradeoff is between time and clutter. For visual menus a balance is sought by simply reducing the amount of information presented or by presenting the information in a more serial manner (for example, breaking up a menu into submenus and presenting them on different screens). The breadth versus depth tradeoff which was discussed earlier in this paper demonstrates this issue as well. For auditory menus the designer can seek a balance by simply reducing the time it takes to convey the sound (e.g. spearcons) or by conveying more information in parallel (playing several sounds at the same time). The parallel sounds approach must be designed carefully to prevent auditory clutter.

Most menus have some textual component which is essential to the purpose of the menu. Although there is significant research involving non-speech sounds to convey text (auditory icons, earcons, etc.), most of them have their drawbacks (as mentioned in this paper). Normal speech might be better to convey the textual message. Spearcons are also very promising, although more research must be conducted to support this. Ideally, a combination of speech and non-speech sounds would probably be the best approach for mobile phone sonification. An example of this would be using spearcons to convey the textual information of each menu item, while earcons or simple beeps are used to provide contextual information and feedback.

Developing guidelines for auditory menus will require innovative thinking in addition to a series of many experiments. A good starting point is to theorize auditory analogies for visual menu concepts. The following are a few suggestions for auditory menu equivalents to visual menu concepts which were discussed earlier in this paper.

### 3.4.1 Content Organization

As mentioned earlier, the breadth versus depth tradeoff defines the way content is organized across screens in a visual menu. Available screen space is usually the limiting factor, and designers try to display as much information as possible without cluttering the screen. In contrast, the design of a purely auditory menu does not rely on screen space at all. Therefore, previous guidelines for the ideal breadth and depth of a menu probably do not apply to an auditory menu. For example, visual menus in a cell phone tend to be deeper and less broad because of the small screen-size. Broader and less deep menus might be more efficient for an auditory interface so that users do not get lost in the deep structure of many levels. On the other hand, due to the serial quality of auditory menus, a menu that is too broad might cause information overload. New experiments must be done to determine guidelines for the breadth versus the depth of a purely auditory menu.

The way content is organized within a single menu frame of an auditory menu might change as well. Consider the ordering of menu items. For instance, if it is logical for the items to be ordered alphabetically in a visual menu, this would probably not change in an auditory menu. However, ordering schemes such as split menus (or ordering by frequency of use) raise several new issues. Since the menu items would most likely be conveyed serially in an auditory menu, sorting by frequency of use has the potential to drastically reduce the time it takes to make a selection. Most of the time, the user would probably only have to listen to the first few items before making a selection. However, further studies have to be done to determine the effect of loss of consistency in this type of auditory menu. Additionally, in a visual split menu, there is usually a visual divider between the split items and regular list so the user knows that they are using a split menu. Auditory split menus would also have to convey this information somehow (perhaps using spatial separation). There are also other ways to reduce the selection time in a serial auditory menu, so that designers might not have to resort to using schemes such as frequency of use or split menus. Time-compression of the speech (like spearcons) would reduce the time needed to listen to the items and, therefore, the selection time. Another method is allowing pre-empting of the sounds. This means that the user can move the focus from one item to another before the sound for that item is done playing. If the user could tell which item was under focus just by listening to the beginning of the sound, they could potentially scroll through all the undesired items very quickly before reaching the target item.

It is possible that experiments will determine that the content organization of an auditory menu should be done very differently from a visual menu. This is fine if the menu is purely auditory. However, if the auditory menu is an enhancement to an existent visual menu, it might be necessary to compromise between the visual and auditory guidelines in order to design the best interaction experience for users of both modalities.

### 3.4.2 Contextual Information

Contextual information includes menu characteristics such as menu size, overall menu structure, and the user's location within the structure. Conveying contextual information is just as important, if not more, for an auditory menu as it is for a visual menu.

*Size of Menu:*

There might be several ways to convey the size of a menu with audio. The TALKS software speaks the item number, followed by the total number of items in the list. For example, if "John Brown" is the tenth contact in the phone's address book, and there are a total of forty-nine contacts, it would speak "John Brown…ten of forty-nine." This information is given for each item in the list. However, the use of non-speech sounds to convey the same information might be more subtle, yet effective. For this same reason, in the visual interface, a scrollbar is used instead of displaying the item number next to each item (subtle, yet effective).

Consider the following sonified version of a mobile phone address book: As the user presses the "down" key on the phone's keypad, the focus shifts from one contact to the next one, and beeps are heard for each contact. The pitches of the beeps are mapped to the location of the menu item (contact). For instance, an item with a low pitched beep might be located lower in the list than an item with a higher pitched beep (or vice versa, depending on what research determines is the most natural polarity). As the user scrolls through the menu items, the pitches of the beeps could give the user an idea of their location within the menu. This method of using pitches to convey location could be considered an *auditory* scrollbar. However, this design does not convey information about the size of the menu. Perhaps the entire range of pitches across the menu could consistently span one whole octave. If the menu consists of ten items, the octave would be divided into ten different, equally spaced pitches. If it consists of fifty items, the octave would be divided into fifty different pitches, etc. Therefore, the difference in pitch between two consecutive items in a long menu would be relatively smaller than in a short menu (analogous to how the thumb of a visual scrollbar is smaller in a long menu and larger in a short menu). Further research should determine whether or not the user would be able to make use of this extra auditory information.

Consider the following alternative approach to designing an auditory scrollbar. For each contact that is under focus, two short beeps are heard: one for the location value of that particular contact, and one for the very last contact in the list. Therefore, the second beep is always at the same pitch and acts as a reference for comparison to the first beep. If the user hears a huge difference in pitch between the two beeps for the first contact, he knows that the list is probably long. As the user scrolls down the list, he hears that the two beeps get closer and closer together in pitch until they are the same (this informs the user that they have reached the end of the list). There are many possible variations of this "two-beep pitch gap" approach: the reference beep could be heard before the location value beep; the reference beep could refer to the very first contact (instead of the last); and so on. These possibilities have not been tested or implemented, but they are ideas that are worth further investigation.

*Location in a Hierarchical Menu Structure:*

The method mentioned above might work well with a linear menu such as an address book. However, a hierarchical menu consists of multiple linear menus in a tree-like structure. In visual menus methods such cascading and the use of background color are used to convey this information. Is it possible to convey this information with audio? Some studies done with earcons (mentioned earlier) seem to be promising in this area. Let us suggest some ideas for this. Previous research on earcons suggests that parallel earcons take up less time than compound earcons. This is consistent with the goal of trying to make auditory menus more parallel than serial. For our purposes, the earcon could consist of simultaneous sounds which can still be distinguished. At any given submenu of a hierarchical menu, a unique earcon would be continuously playing in the background. Consider the following possibilities:

1) Each individual sound that is a part of the earcon refers to an individual root menu that the user had to go through before getting to the current submenu. Therefore, the earcon would keep track of all previous selections. As the user makes another selection, a new submenu opens with a new earcon playing in the background. This earcon sounds the same as the one before it, but it plays the additional sound of the new submenu. In Figure 7 the first menu has a light-gray background color (a background sound). Its submenus have the same sound playing, in addition to another sound (different for each submenu). The third level of submenus has the previous two sounds playing in the background, in addition to an additional unique sound.

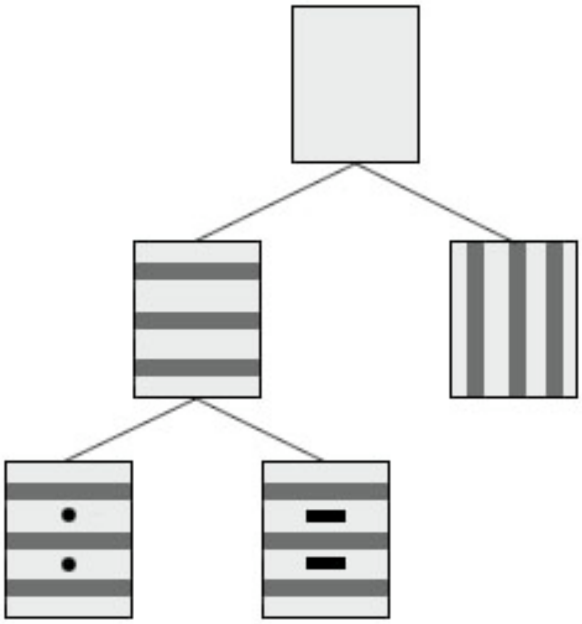

**Figure 7.**

2) A simpler version of the scheme above is one in which the earcons are not unique for each submenu. All horizontally adjacent submenus could have the same background earcon. Therefore, the background earcon would tell the user how deep they are in the hierarchy, but not distinguish between different nodes at the same depth level. See Figure 8.

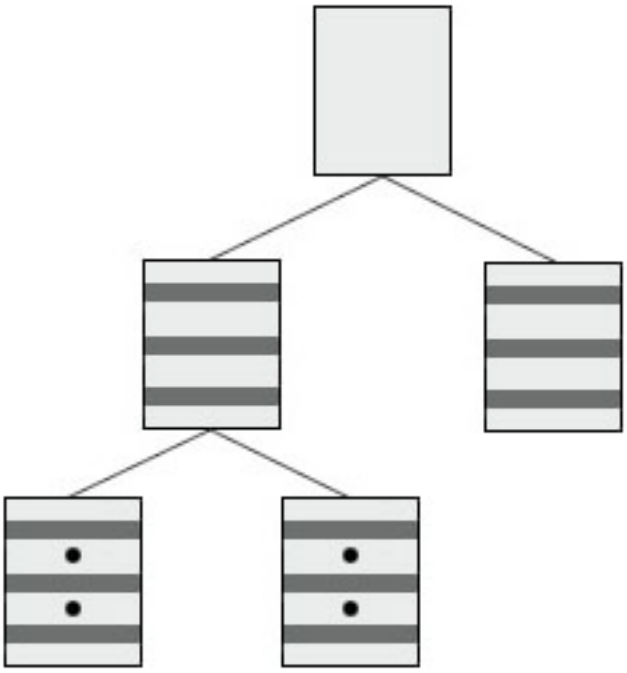

**Figure 8.**

3) An even simpler scheme does not use parallel earcons at all. The background sound is simply unique for each level of the hierarchical structure. This method allows the user to realize that they have entered a different level, but it does not give any information about past selections. See Figure 9.

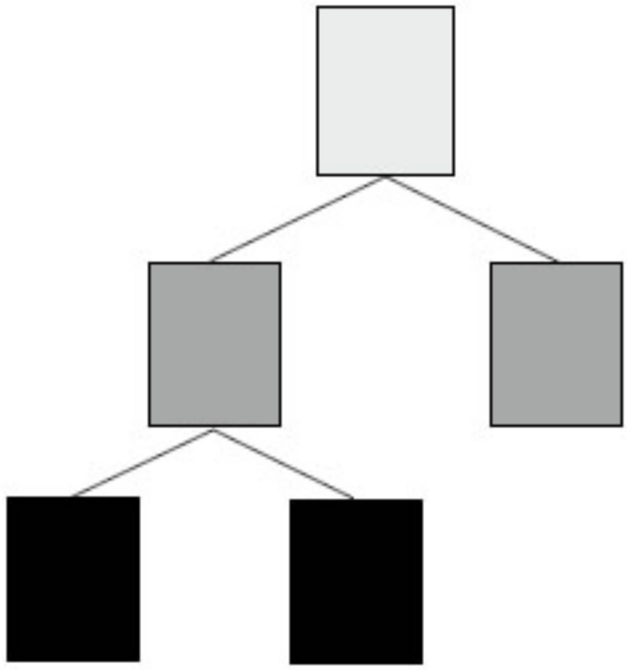

**Figure 9.**

These suggestions are just a subset of many other possibilities. Based on what has already been done, more experiments will have to be devised to determine the best way of using earcons or other sounds (if at all) to convey hierarchical information.

Besides tracking previous selections, it is also possible to predict future outcomes of auditory menus just as it is in a visual menu by indicating what the outcome of a selection will be (before the selection is made). As the user moves the focus of the menu from item to item, a sound could be played to indicate what type of item (leaf, branch, or unavailable) is under focus. One example of a way to sonify this information is the use of different types of text-to-speech. If the menu items are being spoken using text-to-speech, perhaps a male voice could indicate a leaf, a female voice could indicate a branch, and a whisper voice could indicate an unavailable item. As a result, without having selected anything, the user could predict if they have to go deeper in the hierarchy, or if they have arrived at the terminal node for that path.

### 3.4.3 Feedback

As mentioned earlier, non-speech sounds can help give the user feedback during interaction with a menu. Simple beeps can be coordinated with key presses (as is already common in most mobile phones) to confirm the press. Sounds can be different depending on the function. For example, the sound for a focus movement might be a short beep, while the sound for a selection might be a short melody consisting of three notes.

It might be useful to give even more specific feedback about focus movements. As was already mentioned in the previous section, information about what type of item is under focus can be conveyed using distinct sounds for each type. In addition, each different type of movement could sound different. For instance, a T → I movement might sound different from an I → I. A unique sound might be played when wrapping occurs, alerting the user that they have reached the last menu item while scrolling. Similarly, even more specific feedback could be given about the type of selection made. Selection of a leaf item could sound different from selection of a branch item.

If the menu contains shortcuts, their existence and method of invoking them should be made apparent to the user. This can be done by providing auditory feedback (either speech or non-speech) when a focused item has a shortcut associated with it. In the visual menu example for shortcuts, the figure showed a menu which displayed the text "CTRL+N" next to the "New" item to indicate that the user should press the CTRL and N keys to shortcut to that item. In an auditory version of the same menu, text-to-speech could be used to say the shortcut when the item comes under focus.

The table below summarizes some of the visual to auditory analogies discussed above. The "Non-Speech Audio" column simply contains ideas for future research. Most of the ideas have not been tested yet, and it is possible that there are better solutions. Speech was not included as an option in this column because it was assumed that speech is always an option (although it is often not the best option).

**Table 4.** Possible analogies between visual and auditory menu concepts.

| | **VISUAL** | **Non-Speech AUDIO** (speech is always a possible option) |
|---|---|---|
| **Providing Contextual Information** | | |
| *Intrar-frame* | | |
| Number of items | -if # small, already apparent<br>-show total number<br>-scrollbar (size of thumb) | -auditory scrollbar (pitch gap) |
| Location | -highlight<br>-show number<br>-scrollbar (thumb location) | -auditory scrollbar (pitch value) |
| Sort order | -if alphabetic/numeric/sequential, usually apparent<br>-if split menu, show divider<br>-if frequency of use? | -if alphabetic/numeric/sequential, might be apparent<br>-if split, use spatial audio<br>-if frequency of use? |
| *Inter-frame* | | |
| Number of nodes | -usually not shown, necessary? | ? |
| Vertical location (previous selections) | -keep previous root menus visible<br>-show an "address bar"<br>-branch item text = submenu title | -background sounds (hierarchical earcons) |

| | | |
|---|---|---|
| Horizontal location | -show adjacent submenus (cascading menu)<br>-show adjacent submenu titles only (tabs) | -distinct parallel earcons |
| Possible future selections | -show submenu when branch item is focused (not yet selected) | -preview submenu (ex. background speech/sounds) when branch item is focused |
| Possible shortcuts | -at any particular time show all available shortcuts<br>-indicate shortcut to item next to that particular item | -play distinct sound when a focused item has a shortcut |
| **Providing Feedback** | | |
| *Focus move* | -highlight focused item so highlight moves when focus moves | -short beep<br>-spearcon |
| Type of movement (I→I, T→I, wrap…) | -apparent by graphical/textual differences between titles, items, and different submenus | -different sounds for different types of focus moves |
| If item focused, what type (leaf, branch…) | -graphical distinction between item types( ex. show arrow next to branch, gray out unavailable item) | -different sounds for different item types (either for the focus move sound, or in the background) |
| *Selection* | -change the look of focus highlight (ex. color, type, flashing on/off) | -distinct sound (different from focus move sound) |
| Type of selection (branch, procedure) | -branch leads to submenu so visually apparent<br>-procedure leads to the opening of a new window/dialogue box with status information or instructions | -different sounds for different selection types<br>-apparent if branch; focus moves to new submenu with different sound |
| Confirmation after output of selection | -same as above | -happens automatically for branch<br>-auditory status bar for procedure |
| Anticipated point of re-entry | -usually not shown…necessary? | ? |
| **Organizing Content** | | |
| Depth vs. Breadth | -desktop = breadth>depth<br>-cell phone = depth>breadth | -requires more research<br>-might have to stick with visual ratio if multimodal |
| Sort order | -alphabetical, numeric, sequential, frequency of use | -probably same as visual<br>-frequency of use has new implications (auditory is serial, so time important) |
| Movement rules | -varies from menu to menu | -is there one set of rules that is best? |

### 3.4.4 Micro-details

When devising auditory analogies for the visual concepts mentioned above, very comprehensive guidelines need to be developed. For instance, specific ranges for sound parameters have to be determined (pitch, intensity, timbre, etc.). Optimal durations for sounds should be determined, based on the application. Even guidelines for delays between individual sounds should be developed (e.g. what should the delay between compound earcons in a sonified mobile phone be?).

For speech, parameters such as frequency, male or female voice, and speed of speech should be determined. If the speech is to be time-compressed, there are many techniques which should be considered. Various speedup (linear, speech rate normalized, increasing within a phrase, etc.) and excision (removing silences or unnecessary utterances) techniques have already been compared, and the excision techniques were found to be more effective (Tucker & Whittaker 2006). These results might not be consistent for the design of spearcons, which are not necessarily meant to be intelligible. Thus, further studies should be conducted to determine the best spearcon generating technique.

## 3.5 Auditory Menus outside the Context of Visual Menus

It is often difficult to think of designing an auditory menu without basing it on our knowledge of visual menus, which helps guide us in developing the necessary components of auditory menus. Although they are each developed differently, auditory icons, earcons, and spearcons are all simply audio versions of visual menu items (discussed in the first part of this report). This is a good starting point, but eventually researchers and designers should be able to "think outside of the box" and develop novel guidelines and designs for auditory menus. These innovative designs might be more suitable for auditory menus, and they might not even have a direct translation for visual menus. Specifically, it is important to reach this stage so that visually impaired users can take advantage of the full potential of auditory menus.

In addition, many of the results mentioned thus far imply that multi-modal interaction is better than using only visual or auditory means. However, Bussemakers and de Haan (1998) found that when the "primary message is presented visually, any accessory information that is provided in the auditory modality causes a delay." Conversely, when the primary message is "presented auditorily, in all cases the accessory visual information speeds up the response." Therefore, the authors suggest that great care must be taken when designing a multi-modal interface that uses sound as an accessory to visual information.

# 4 References

Asakawa, C., Takagi, H., Ino, S., & Ifukube, T. (2003). Maximum listening speeds for the blind. Proceedings ICAD ’03, June 6-9, Boston, Massachusetts, United States.

Bederson, B. (2000). Fisheye menus. Proceedings of the 13th annual ACM symposium on User interface software and technology, p.217-225, November 06-08, 2000, San Diego, California, United States.

Blattner, M., Sumikawa, D. & Greenberg, R. (1989). Earcons and icons: Their structure and common design principles. Human Computer Interaction 4(1): 11-44.

Brewster, S.A., Wright, P.C. & Edwards, A.D.N. (1992). A detailed investigation into the effectiveness of earcons. In G. Kramer (Ed.), Auditory display, sonification, audification and auditory interfaces. The Proceedings of the First International Conference on Auditory Display, Santa Fe Institute, Santa Fe, NM: Addison-Wesley, pp. 471-498.

Brewster, S.A., Wright, P.C. & Edwards, A.D.N. (1995). Parallel earcons: Reducing the length of audio messages. International Journal of Human-Computer Studies, 43(2), pp. 153-175.

Brewster, S.A., Wright, P.C. & Edwards, A.D.N. (1995). Experimentally derived guidelines for the creation of earcons. In Adjunct Proceedings of HCI'95, Huddersfield, UK.

Brewster, S.A. (1996). The design of a sonically-enhanced interface toolkit. Department of Computing Science Technical Report TR-1996-23, June.

Brewster, S.A. Raty, V.-P. & Kortekangas, A. (1996). Earcons as a Method of Providing Navigational Cues in a Menu Hierarchy. In Proceedings of HCI'96 (Imperial College, London, UK), Springer, pp 167-183.

Brewster, S.A. (1997). Navigating telephone-based interfaces with earcons. In Proceedings of BCS HCI'97 (Bristol, UK), Springer Verlag, pp 39-56.

Brewster, S.A., Capriotti, A., Hall, C.V. (1998). Using compound earcons to represent hierarchies. HCI Letters, 1(1), pp 6-8.

Brewster, S.A., Leplatre, G. and Crease, M.G. (1998). Using Non-Speech Sounds in Mobile Computing Devices. In Johnson C. (Ed.) Proceedings of the First Workshop on Human Computer Interaction with Mobile Devices, (Glasgow, UK), Department of Computing Science, University of Glasgow, pp 26-29.

Brewster, S.A. and Crease, M.G. (1999) Correcting Menu Usability Problems With Sound. Behaviour and Information Technology 18(3), 165-177.

Brown, J. W. (1982). "Controlling the complexity of menu networks." Communications of the ACM. 25(7):412-418.

Bussemakers, M.P., & De Haan, A. (1998). Using Earcons and Icons in Categorisation Tasks to Improve Multimedia Interfaces. In Proceedings of ICAD'98 (pp. xx). Glasgow, UK: British Computer Soc.

Byrne, M.D., Anderson, J.R., Douglass, S., & Matessa, M. (1999). Eye tracking the visual search of click-down menus, Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit, p.402-409, May 15-20, 1999, Pittsburgh, Pennsylvania, United States.

Card, S. K. (1982). User perceptual mechanisms in the search of computer command menus. In Human Factors in Computer Systems Proceedings. ACM, New York, 190-196.

Chae, Minhee, and Kim Jinwoo. Do size and structure matter to mobile users? An empirical study of the effects of screen size, information structure, and task complexity on user activities with standard web phones. Behavior and Information Technology, May-June 2004, VOL. 23, NO. 3, 165–181

Cherry, E.C. (1953). Some experiments on the recognition of speech, with one and two ears. J. Acoust. Soc. Am., vol. 25, pp. 975.

Dix, A., Finlay, J., Abowd, G., Beale, R. (1998) Human-Computer Interaction. 2nd Ed. Prentice Hall Europe.

Gaver, W. W. (1986). Auditory icons: Using sound in computer interfaces. Human-Computer Interaction. 2, 167 - 177.

Geven, Arjan, Sefelin, Reinhard, and Tscheligi, Manfred. (2006). Depth and Breadth away from the Desktop – the Optimal Information Hierarchy for Mobile Use. ACM International Conference Proceeding Series; Vol. 159, pp.157-164.

Hornof, A. J., & Kieras, D. E. (1997). Cognitive modeling reveals menu search is both random and systematic. Proceedings of CHI '97, New York: ACM, 107-114.

Kiger, J.I. (1984). The depth/breadth trade-off in the design of menu-driven user interfaces. International Journal of Man-Machine Studies, 20, 201-213.

Leplatre, G. and Brewster, S.A. (2000). Designing Non-Speech Sounds to Support Navigation in Mobile Phone Menus. In Proceedings of ICAD2000 (Atlanta, USA), ICAD, pp 190-199.

Lorho, G., Marila, J., Hiipakka, J. (2001), Feasibility of Multiple Non-Speech Sounds Presentation Using Headphones, in Proc. of ICAD'01, Espoo, Finland, 32–37.

Lorho, G., Marila, J., Hiipakka, J. (2002), Structured Menu Presentation Using Spatial Sound Separation Mobile HCI 2002, LNCS 2411, pp. 419-424.

Marila, J. (2002). Experimental comparison of complex and simple sounds in menu and hierarchy sonification. Proceedings of the International Conference on Auditory Display (pp. 104-108), Kyoto, Japan.

Miller, D.P. (1981). The depth/breadth trade-off in hierarchical computer menus. In Proceedings of the Human Factors Society, 25th Annual Meeting, 296-300.

Mitchell, J. AND Shneiderman, B. (1989). Dynamic versus static menus: An exploratory comparison. SIGCHI Bull. 20, 4, 33-37.

Norman, Kent (1991). The Psychology of Menu Selection: Designing Cognitive Control at the Human/Computer Interface. http://www.lap.umd.edu/poms/

Palladino, D., & Walker, B. N. (2007). Learning rates for auditory menus enhanced with spearcons versus earcons. Proceedings of the International Conference on Auditory Display (ICAD 2007), Montreal, Canada (26-29 June). pp. 274-279.

Perlman, Gary, & Sherwin, Leo C., (1988). Designing menu display format to match input device format, ACM SIGCHI Bulletin, v.20 n.2, p.78-82, Oct. 1988.

Sears, Andrew, and Shneiderman, Ben (1994). Split Menus: Effectively Using Selection Frequency to Organize Menus. In ACM Transactions on Computer-Human Interaction (TOCHI), p. 27-51.

Shneiderman, B. (1986). Designing menu selection systems. Journal of the American Society for Information Science, 37(2), 57-70.

Snowberry, K., Parkinson, S. and Sisson, N. (1983), Computer display menus. Ergonomics, 26(7), 699 – 712.

Teitelbaum, R. C.; and Granda, R. The effects of positional constancy on searching menus for information. In: Proceedings of CHI '83: Human Factors in

Tucker, S. and Whittaker, S. (2006). Time is of the essence: an evaluation of temporal compression algorithms. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Montreal, Canada (22-27 April).

Walker, V. A. and Brewster, S. A. (2000). Spatial audio in small screen device displays. Personal Technologies 4, 2, 144--154.

Walker, B. N., Nance, A., & Lindsay, J. (2006). Spearcons: Speech-based Earcons Improve Navigation Performance in Auditory Menus. Proceedings of the International Conference on Auditory Display (ICAD 2006), London, England (20-24 June). pp. 63-68.