Chapter 6:

AUDITORY ICONS

Introduction

An account of everyday listening, as described in the last chapter, gives us a new foundation for creating auditory interfaces. Instead of mapping information to sounds and their dimensions, as the interfaces we described in Chapters 3 and 4 do, we can encode information using audible events and their dimensions. The results are auditory icons, everyday sounds designed to convey information about events by analogy to everyday sound-producing events.

For instance, when we mark a computer file for deletion, we might hear the sound of an object crashing into a wastebasket. The type of object we hear might indicate the type of object we've discarded – whether it is a text file, a program, etc. – while its size might be indicated by the size of the object we hear. The number of other objects awaiting deletion might effect the sound of the crash, and their types and sizes could also be indicated. Finally, when the files are eventually deleted, we might hear an appropriate sound (for instance, a trash compactor) as the deletion occurs, or, better, enough in advance that we can abort the deletion.

Auditory icons such as these are like sound-effects for computers, complementing visual events with appropriate sounds. But they are not designed merely to provide entertainment, rather they convey rich information about events in computer systems, allowing us to listen to computers as we do to the everyday world. They use a strategy similar to that used in creating visual icons, mapping events in computer systems to related everyday events in order to aid learning and comprehension. This strategy contrasts with those based on

musical listening. For instance, compare the
"wastebasket" sound of our example with the simple two-note "earcon" for indicating file deletion proposed by Sumikawa et al. (1988?), as described in Chapter 4. Not only does the wastebasket sound convey more information about the event and objects involved in it, but it does so in a more intuitive way, and one which fits better with the graphical components of the interface.

The richness of everyday sounds, their ease of comprehension, and their ability to help create a virtual world, lead to a number of powerful advantages for auditory icons. In this chapter, we review work on auditory icons, describing how they may be created and some of the issues to keep in mind when doing so. A number of systems that use auditory icons are described, both to give an idea of the kinds of auditory icons that might be created and to illustrate the range of functions they can perform. A number of issues that are raised by this work are discussed, and their importance for all auditory interfaces is considered. Finally, we discuss the design and implementation of systems that use auditory icons.

Advantages of Auditory Icons

As the example above suggests, auditory icons are simply everyday sounds mapped to events in the computer. So marking something as deleted is expressed by the sound of throwing it into a trashcan. Selecting an object might sound like touching it. Moving an object might make a scraping sound.

But auditory icons don't serve merely as labels for categories of events and objects, as most visual icons do. They can be *parameterized* to reflect their relevant dimensions as well (see Figure 6.1). To return to the example of deleting a file, the computer event (deletion) is mapped to an everyday event (throwing an object into a trashcan). Then parameters of the everyday event can be used to indicate parameters of the computer event: the size of the object represents the size of the computer file, its material reflects the file type, etc.

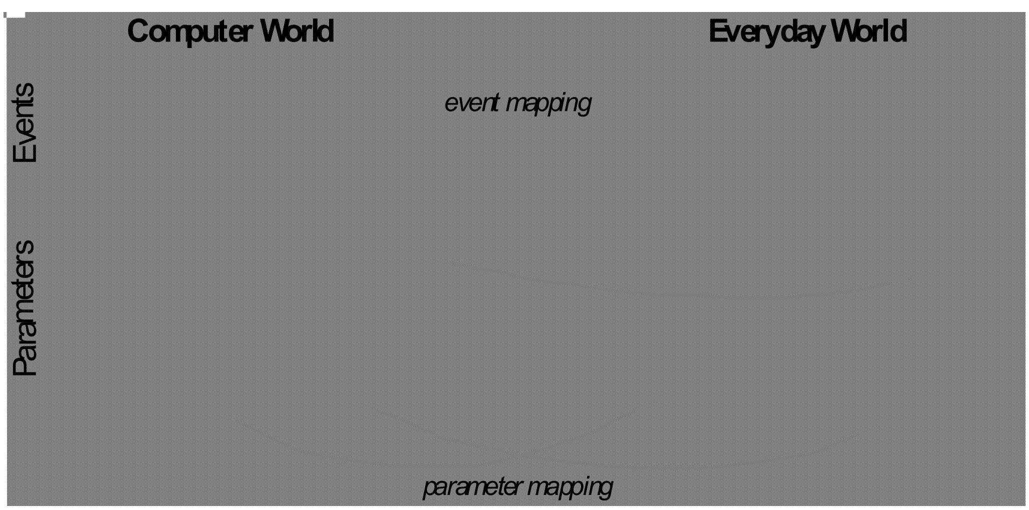

Figure 6.1. Auditory icons are created by mapping events in the computer world to sound-producing events in the everyday world. They may be parameterized by mapping parameters of the computer events to dimensions of the everyday event that affect the sound it makes.

Thus auditory icons convey multidimensional information by mapping the dimensions of the information to be displayed to dimensions of everyday events. In this way, any one sound can convey a great deal of information. Moreover, "families" of auditory icons can be created by exploiting the organization inherent in everyday events. For instance, if the material of a sound-producing event is used to represent the type of object, all auditory icons concerning that type of object would use sounds made by that kind of material. So text files might always sound wooden, whether they are selected, moved, copied or deleted. In this way, a rich system of auditory icons may be created that relies on relatively few underlying metaphors.

Because auditory icons rely on everyday sounds, it is relatively easy to make them compatible with existing graphic interfaces. Objects can sound like what they look like: files can sound like solid objects being tapped or scraped, windows can sound like glass surfaces, objects clatter when they are thrown away. Achieving this correspondence between sound and graphics is more difficult using the strategies described in Chapter 4 because they rely on musical parameters of sound. The earcon for deleting a file described in Chapter 4, for instance, bears little relation to the graphic counterpart of dropping a file icon over a trashcan icon. A different set of rules must be learned for the auditory and graphic components of such systems. Using auditory icons, in contrast, a single set of metaphors can guide both aspects of the interface.

When the same analogy underlies both auditory and visual icons, the increased redundancy of the interface can help users learn and remember the system. In addition, making the model world of the computer consistent in its visual and auditory aspects increases users' feelings of *direct engagement* (Hutchins, Hollan, & Norman, 1986) or *mimesis* (Laurel, 1986) with that world. The concepts of direct engagement and mimesis refer to the feeling of working in the world of the task, not the computer. By making the model world of the computer more real, one makes the existence of an interface to that world less noticeable. Providing auditory information that is consistent with visual feedback is one way of making the model world more vivid. In addition, using auditory icons may allow more consistent model worlds to be developed, because some computer events may map more readily to sound-producing events than to visual ones.

In sum, the strategy of creating auditory icons by mapping sound-producing events to events in the computer has many useful features. It allows the creation of parameterized auditory icons that convey rich multidimensional information. Families of auditory icons can be designed that relate to one another in systematic ways. Consistent auditory and graphical mappings can be created, producing a coherent model world that increases the transparency of the interface.

In the next sections we describe a variety of interfaces that have used auditory icons as a way of illustrating these points. In addition, these examples stress the variety of functions that auditory icons – and sounds in general – may perform in the interface. Examples such as these are the best way to communicate the concept of auditory icons and to explore their creation and functionality.

Systems Which Use Auditory Icons

The benefits of auditory icons can be more fully illustrated by examples of systems that have used them. A number of systems have been created which illustrate the potential for auditory icons to convey useful information about computer events. In particular, these systems suggest that sound is well suited for providing information:

- about previous and possible interactions,
- indicating ongoing processes and modes,
- useful for navigation, and
- to support collaboration.

In the following sections, a variety of systems that have used auditory icons are briefly described. Their order is roughly chronological and, not surprisingly, also reveals the increasing functionality that auditory icons can provide.

The SonicFinder: Creating an Auditory Desktop

The SonicFinder (Gaver, 1989) is the first interface to incorporate auditory icons. Developed for Apple Computer Company, it is an extension to the Finder, the application used to organize, manipulate, create and delete files on the Macintosh. The Finder is automatically run when the machine is booted, and thus is probably the program most frequently encountered by Macintosh users. Because of this, and because the SonicFinder was easily portable (requiring no special hardware, it could be distributed on a single 800K floppy disk), many people have encountered the SonicFinder and have provided useful feedback about its utility.

Creating the SonicFinder required extending the Finder code at appropriate points to play sampled sounds modified according to attributes of the relevant events. Thus a variety of actions make sound in the SonicFinder: selecting, dragging, and copying files; opening and closing folders; selecting, scrolling, and resizing windows; and dropping files into and emptying the trashcan. Most of these sounds are parameterized, although the ability to modify sounds is limited. So, for instance, sounds which involve objects such as files or folders not only indicate basic events such as selection or copying, but also the object's types and sizes via the material and size of the virtual sound-producing objects. In addition, the SonicFinder incorporates an early example of an *auditory process monitor* in the form of a pouring sound that accompanied copying and that indicates, via changes of pitch, the percentage of copying that had been completed (c.f. Cohen, this volume).

Because the Macintosh is a single-processing machine with a fairly simple interface, the sounds used in the SonicFinder basically provide feedback and information about possible interactions (as well as more general information about file size and type, dragging location, and the like). Nonetheless, it provides a valuable example of the potential of auditory icons, showing that sounds such as these can be incorporated in an intuitive and informative way.

Apple has never released the SonicFinder (although it did appear on a developer's CD-ROM). There are several reasons for this. Most fundamentally, the perceived benefit to disk-space ratio was not high enough: because it uses sampled sounds, the SonicFinder could never be reduced below about 100K in size – prohibitively large to release in the days before high-density floppy disks and CD-ROMs. Although many people found the auditory cues useful, others found them irritating or thought of them as merely entertaining. This provides a valuable example of the real-world challenges that designers of auditory interfaces must face. Nonetheless, the interface has spread around the world in what Buxton has termed the "research underground," and has hopefully helped to demonstrate the potential and appeal of auditory interfaces in general.

SoundShark: Sounds in a Large Collaborative Environment

Although the SonicFinder is useful in incorporating auditory icons into a well-known and often-used interface, its simplicity may lead people to underestimate the functions that auditory icons may serve. For this reason, Gaver and Smith (1991) demonstrated auditory icons used in a large-scale, multiprocessing, collaborative system called SharedARK, and dubbed the resulting auditory interface SoundShark.

SharedARK is a collaborative version of ARK, the Alternate Reality Kit. Developed by Smith (1989), ARK is designed as a virtual physics laboratory for distance education. The "world" appears on the screen as a flat surface on which a number of 2.5D objects may be found. These objects may be picked up, carried, and even thrown using a mouse-controlled "hand." They may be linked to one another, and messages may be passed to them using "buttons." Using this system, a number of simple physical experiments may be performed. In addition, SharedARK allows the same world to be seen by a number of different people on their own computer screens (and is usually used in conjunction with audio and video links that allow them to see and talk to one another). They may see each other's hands, manipulate objects together, and thus collaborate within this virtual world.

SharedARK is a multiprocessing system, with the potential for several "machines" or self-sustaining processes to run simultaneously. In addition, it provides a very large world to users, in that the space for interaction is many times larger than the screen (depending on available memory, it may cover literally acres of virtual space). Users move around this space by moving their hand near the edge of the window, causing it to scroll over adjacent territory. To help with navigation, a "radar view" is presented which shows a much-reduced representation of the world and objects within it.

This interface was extended by adding auditory icons to indicate user interactions, ongoing processes and modes, to help with navigation, and to provide information about other users. Sounds were used to provide feedback as they were in the SonicFinder: Many user actions were accompanied by auditory icons which were parameterized to indicate attributes such as the size of relevant objects. In addition, ongoing processes made sounds that indicated their nature and continuing activity even if they were not visible on the screen. Modes of the system, such as the activation of "motion," which allows objects to move if they have a velocity, were indicated by low-volume, smooth background sounds. Collaborators could hear each other even if they couldn't see each other, which seemed to aid in coordination. Finally, distance between a given user's hand and the source of the sound was indicated by the sounds' amplitude and by low-pass filtering, aiding with navigation. The apparent success of this manipulation led us to develop "auditory landmarks," objects whose soul function was to play a repetitive sound that could aid orientation.

SoundShark was implemented using an external sampler that was triggered and controlled via MIDI. This allowed a number of features that would have been more difficult to achieve had the sound playback been handled by the workstation. Multiple sounds could be played simultaneously, and manipulations such as varied attack times and low-pass filtering could be used. Nonetheless, the use of external hardware meant that the system was not as portable as the SonicFinder.

ARKola: Studying the Use of Sound in a Complex System

Our experiences with SoundShark suggested that auditory icons could provide useful information about user-initiated events, processes and modes, and about location within a complex environment. To test this, we developed a special application within SoundShark which we used as a basis for observing people's use of the system. This application, developed in collaboration with Tim O'Shea, was a model of a softdrink plant called the ARKola bottling factory (Gaver et al, 1992). It consisted of an assembly line of 9 machines which cooked, bottled, and capped cola, provided supplies, and kept track of financing. The plant was deliberately designed to be too large to fit on the computer screen, so participants could only see about half the machines at any given time. In addition, we designed the plant to be fairly difficult to run, with the rates of the machines requiring fine tuning and with machines occasionally "breaking down," necessitating the use of a "repair" button.

Each of the machines made sounds to indicate their function. For instance, the "nut dispenser" made wooden impact sounds each time a nut was delivered to the cooker, the "heater" made a whooshing flame-like sound, the "bottler" clanged and the "capper" clanked. In addition, the rate of each machine was indicated by the rate of repetition of the sounds it made, and problems with the machines were indicated by a variety of alerting sounds such as breaking glass, overflowing liquid, and so forth.

With as many as 12 sound playing simultaneously, designing the sounds so that all could be heard and identified was a serious challenge. In general, we used temporally complex sounds to maximize discriminability, and designed the sounds to be semantically related to the events they represented. Two strategies were found to be useful in avoiding masking. First, sounds were spread fairly evenly in frequency, so that some were high-pitched and others lower. Second, we avoided playing sounds continuously and instead played repetitive streams of sounds, thus maximizing the chance for other sounds to be heard in the gaps between repetitions.

Six pairs of participants were asked to run the plant with the aim of making as much "money" as they could during an hour-long session. Each pair ran the plant for two hours, one with and one without auditory feedback (with the order, of course, being counterbalanced). We observed their performance from a "control room" via video links as they ran the plant, and videoed their activities for later analysis.

Our observations indicated that sounds were effective in two broad areas. First, they seemed to help people keep track of the many ongoing processes. The sounds allowed people to track the activity, rate, and functioning of normally running machines. Without sound, people often overlooked machines that were broken or that were not receiving enough supplies; with sound these problems were indicated either by the machine's sound ceasing (which was often ineffective) or by the various alert sounds. Perhaps most interesting, the auditory icons allowed people to hear the plant as an integrated complex process. The sounds merged together to produce an auditory texture, much as the many sounds that make up the sound of an automobile do. Participants seemed to be sensitive to the overall texture of the factory sound, referring to "the factory" more often than they did without sound.

The second set of observation related to the role of sound in collaboration. In both the sound and no-sound conditions, participants tended to divide responsibility for the plant so that each could keep one area on the screen at all times. Without sound, this meant that a each had to rely on their partner's reports to tell what was happening in the invisible part. With sound, each could hear directly the status of the remote half of the plant. This seemed to lead to greater collaboration between partners, with each pointing out problems to the other, discussing problems, and so forth. The ability to provide foreground information visually and background information using sound seemed to allow people to concentrate on their own tasks, while coordinating with their partners about theirs.

Sound also seemed to add to the tangibility of the plant and increased participants' engagement with the task. This became most evident when one of a pair of participants who had completed an hour with sound and were working an hour without remarked "we could always make the noises ourselves..." In sum, the ARKola study indicated that auditory icons could be useful in helping people collaborate on a difficult task involving a large-scale complex system, and that the addition of sounds increased their enjoyment as well.

ShareMon: Background Sounds for Awareness

As we have pointed out earlier, continuous or repetitive sounds have the interesting property that they fade to the background of attention over time. They become part of the peripheral information that people have access to, but don't consciously attend to at a given time. People *can* bring them into the focus of attention, and when the sounds change in some way they are likely to force a switch of attention, but when they remain relatively unchanging they are relatively unnoticeable.

Continuous and repetitive sounds thus have great potential for use as background status indicators. Both SoundShark and the ARKola factory used continuous sounds to help people keep track of ongoing processes and modes. Some attempt was made to keep these sounds unobtrusive, but for the most part they were about as noticeable as any of the other sounds used in this system. Though these systems began to explore the potential of sounds used as unobtrusive status indicators, much of the potential for using sound in this way remained unexplored.

Recently Jonathan Cohen of Apple Computer has been investigating the use of sounds for maintaining background awareness in more detail (Cohen, 1993).

• • •

EAR: Environmental Audio Reminders

Where SoundShark and ARKola explored the use of auditory icons to support collaboration in software systems, another system, called EAR (for Environmental Audio Reminders), demonstrates that auditory icons are also helpful for supporting collaboration in the office environment itself (Gaver 1991). This system plays a variety of nonspeech audio cues to offices and common areas inside EuroPARC to keep us informed about a variety of events around the building. It is one element of ongoing research at EuroPARC on *environmental interfaces*, which are aimed at merging the power of the computational and everyday environments. EAR works in conjunction with the RAVE audio-video network (Gaver et al., 1992; Buxton & Moran, 1990), which connects all the offices at EuroPARC with audio and video technologies using a computer-controlled switch, and *Khronika* (Lövstrand, 1991), an event server which uses a database of events in conjunction with software daemons to inform us of a wide range of planned and spontaneous, electronic and professional events. EAR, then, consists of sounds triggered by Khronika when relevant events occur, which are routed using the RAVE system from a central server (in our case, a Sparcstation) to any office in the building.

A wide variety of sounds are used to remind us about a range of events. For instance, when new email arrives, the sound of a stack of papers falling on the floor is heard. When somebody connects to my video camera, the sound of an opening door is heard just before the connection is made, and the sound of a closing door just after the connection is broken. Ten minutes before a meeting, the sound of murmuring voices slowly increasing in number and volume is played to my office, then the sound of a gavel. And finally, when one of my colleagues decides to call it a day, they often play the "pub call" to my office, the sound of laughing, chatting voices in the background with the sound of a pint glass being filled with real ale in the foreground.

Many of the sounds we use in EAR may seem frivolous because they are cartoon-like stereotypes of naturally-occurring sounds. But it is precisely *because* they are stereotyped sounds that they are effective. More "serious" sounds – such as electronic beeps or sequences of tones – would be likely to be less easily remembered than these. In addition, we have taken some care in shaping the sounds to be unobtrusive. For instance, many of the sounds are very short; those that are longer have a relatively slow attack so that they enter the auditory ambience of the office subtly. Most of the sounds have relatively little high-frequency energy, and we try to avoid extremely noisy or abrupt sounds. So though the sounds we use are stereotypes, they are designed to fit into the existing office ambience rather than intruding upon it.

In sum, the auditory cues used in the EAR system can be unobtrusive, informative, and valuable. They serve to indicate events in the same way that they might be heard in everyday life, with the added advantage that the events cued are chosen by users. They allow us to hear distant events, or events that don't naturally produce informative noises, helping to blur the distinction between the electronic and physical environments. By informing us about ongoing events in the building they help to ease the transition between working alone and working together.

Summary

These systems demonstrate the wide range of functions that auditory icons can perform. They can provide information about user actions, about possibilities for new actions, and about non-visible attributes of objects in the system. They can provide background information about processes and modes in more complex system. Continuous or repetitive sounds which are varied according to distance may serve as auditory landmarks, supporting navigation in complex systems. Finally, auditory icons can work with graphic displays, supporting a smooth flow between individual and cooperative work.

These examples also demonstrate the range of systems which may benefit from auditory icons, from traditional desktop graphical interfaces to more complex virtual realities and process simulations, to systems which introduce computational power into the everyday environment itself. By building these systems, using them ourselves, and observing others use them, we have gained a great deal of valuable information about their utility, their problems, and issues for their design.

Finally, these systems illustrate the broad range of sounds that may be used as auditory icons, from the simple impact and scraping sounds that are designed for graphical user interfaces such as the SonicFinder, to the more complex and continuous process sounds used in SoundShark, ARKola, and ShareMon, to the very complex and stereotypical sounds used in EAR. They also illustrate some of the issues that must be tackled when designing sounds for interfaces, particularly the tension inherent in designing sounds that are simultaneously identifiable and subtle, memorable but not annoying. The rest of this chapter focuses on the creation of such auditory icons, discussing some of the issues that are important for the creation of auditory icons, and, more generally, for the creation of any auditory interface.

Issues for Auditory Icons

A number of issues become clear in considering the interfaces described above. These include questions regarding auditory icon's functions, mapping, vocabulary, and annoyance:

- *Functions*: What are auditory icons good for? We have addressed this already, through the examples we have described.
- *Mapping*: How should sounds be mapped to events? The different sorts of mapping from perceptible representations to underlying events.
- *Vocabulary*: What sounds should be used? Is hearing a telephone an example of everyday listening? How can sounds be both recognizable and discriminable?
- *Annoyance*: How can we use sounds without driving users crazy?

In the rest of this chapter, we consider each of these issues in turn – with the exception of functions, which we have discussed in the previous sections on examples of systems that use auditory icons. In addition, we discuss practical issues concerning the implementation of auditory icons in the next chapter of this book.

Mapping Sounds to Events

The two defining features of auditory icons are the use of environmental sounds controlled along parameters of the events which cause them, and the use of intuitive mappings between the sounds and the computer events they indicate.

In fact, the type of mapping has priority over the sounds used. It is entirely possible to use environmental sounds in a way that is incompatible with the concept of auditory icons. For instance, Brown et al. (1989) report an experiment in which they showed that everyday sounds were as effective at facilitating a spatial search task as graphical cues were. In order to indicate the six columns of their visual display, they used applause sounds differentiated by the number of people clapping (1 or 3) and the repetition rate (from 50 to 208 bpm). This mapping between applause and spatial location is extremely arbitrary – a fact which the authors recognized, pointing out that any mapping between everyday sounds and spatial location is liable to be arbitrary. So while this is an impressive demonstration of the utility of nonspeech audio, it is a poor example of the use of auditory icons.

It is easy to say that the mapping between an auditory icon and the event it represents should be intuitive, but more difficult to know what makes a particular mapping intuitive. In order to understand this, it is necessary to discuss what is being mapped to what, and the variety of mappings that might occur[1].

What is Being Mapped to What?

Consider what happens when we throw a computer file away (Figure 2). There are several ways to consider what is going on. At the level of the display, we are simply throwing the file icon into that of the trashcan, knowing the file icon will disappear when the trash is "emptied."

But of course the purpose is not just to get rid of the icon, but also the file itself. We are happy with throwing the icon away because it stands for the real file, it is the perceptual reality of the file. Throwing its icon becomes equivalent to throwing a real file into the trash; not only does its apparition go away, but the file itself goes away with all the attendant consequences (e.g., it is no longer accessible for reading, copying, etc.). The display is the outward appearance of the model world of files and wastebaskets set up by the computer, a model world which not only determines the current display, but also future possibilities for action.

Still, in reality we aren't throwing away a file at all. Instead we are manipulating a graphical output, which is linked to data structures in the computer. What we are *really* doing is something like switching an address bit from one to zero, or... The point is that we don't need to know. The reality of the computer is simultaneously hidden and made accessible to us by the conceptual mapping that allows electronics to be thought of as information, information to be thought of as variables, variables to be grouped into structures, and structures to be called "files".

---

[1] This issue has been discussed in more detail by Gaver (1988), and more recently by xxx. Understanding the relations between representations and the things they represent is the topic of semiotics; for more general readings in this field see, for instance, xxx.

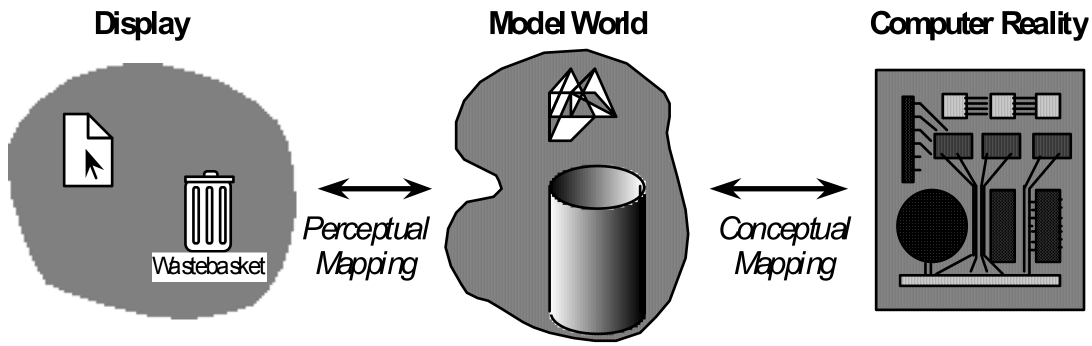

Figure 2: There are two sorts of mappings, conceptual and perceptual, between the display the user sees and the reality of the computer. Conceptual mappings are always metaphorical abstractions, but a variety of perceptual mappings may be used.

In light of this example, it is clear that there are two sorts of mappings that mediate between throwing a file icon away and the actual physical events that occur in the computer. The first is the *conceptual mapping* that allows the electronic events to be thought of in terms of everyday analogies. The second is the *perceptual mapping* allowing the model world set up by these analogies to be expressed in terms of perceptible entities such as icons (or sounds).

Types of Mapping

Conceptual mappings between the reality of silicon gates and the like and the model world of files, trashcans and so on are always metaphorical (Hutchins, 1987). But there are three different kinds of mappings that may allow the model world to be expressed as a display: *symbolic, metaphorical,* and *iconic*. Figure 3 shows examples of the three for both visual and auditory displays showing that a file has been marked for deletion. The first kind of mapping is symbolic: an arbitrary sign is used to mark the file – an X for the visual display, and a simple beep for the auditory one. The second kind of mapping is metaphorical: the displays form an analogy to the real-world metaphor of deletion. For the visual display, this is accomplished by fading the icon, as if it were in the process of disappearing. For the auditory display, an earcon is used in which a motive standing for a file is diminished in amplitude[2]. Finally, the third kind of mapping is iconic: the display resembles the real-world event in a lawful way. In the graphical version, an icon of the file is dropped over the icon of the trashcan. For the auditory display, the sound of an object dropping into a (full) trashcan is played.

[2] Note that the mapping between the file and the motive in this example is wholly symbolic in that the choice of motive is arbitrary. The metaphor here is between the deletion of the file and the decreasing volume of the motive.

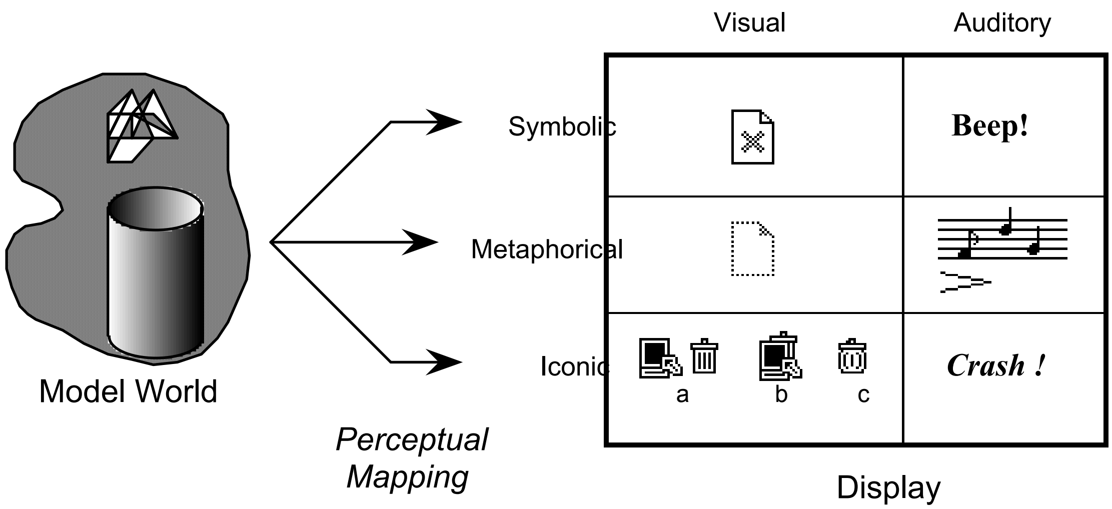

Figure 3. There are three types of mappings used between the model world of the computer and the display: symbolic, metaphorical, and iconic.

Symbolic, metaphorical, and iconic mappings are distinguished by the degree to which they are arbitrary or lawful; this also has profound effects on the ease with which they are learned and remembered. Symbolic mappings are difficult to learn because they are entirely arbitrary and rely on social convention for their establishment. Metaphorical mappings are easier to learn because they rely on similarities between the representation and the thing to be represented. Iconic mappings are easiest to learn because they rely on the similarity between a representation and the thing it represents: in the case of perceptual mappings, this means that the representation must look like (or sound like) the event it represents. Note that this similarity is not a subjective quantity, but rather depends on the lawful, physical mapping between physical structure and perceptible appearance in the everyday world. Iconic mappings allow learners to apply the skills they already possess in the everyday world to the new model worlds created by computer technology.

From this point of view, designers of auditory interfaces should be concerned with using mappings between sounds and events that are as lawful as possible. In other words, they should avoid creating new symbolic languages for sounds, and instead seek iconic or metaphorical mappings. This is one of the key concerns behind the concept of auditory icons.

The Vocabulary of Auditory Icons

At the beginning of this chapter, we defined auditory icons as everyday sounds that convey information about computer events by analogy with everyday events. As the discussion above emphasizes, it is the mapping between sound and event that is crucial for auditory icons – auditory icons could be constructed from musical tones or motives, if they mapped in an iconic or strongly metaphorical way to their intended message. From this point of view, the use of everyday sounds for auditory icons may seem accidental and unnecessary.

But the fact that most auditory icons use everyday sounds is no accident. Instead, it is precisely because everyday sounds are defined in terms of events that lawful mappings can be created between auditory icons and the events they convey. The new framework for sound offered by a study of everyday listening, as outlined in Chapter 5, serves as a new palette for designers of auditory interfaces. Instead of trying to map dimensions of a computer event to dimensions of sound, one can now map events to events. This offers the possibility of the sort of lawful, iconic and metaphorical mappings illustrated by the systems described in this chapter.

Thus most auditory icons should use everyday sounds. If a file is selected by clicking the mouse over it, then a tapping sound should be produced. If it is moved by dragging it over a background, then a scraping sound provides appropriate feedback. Whenever possible, an iconic mapping between the event in the model world and the sound that is heard should be used.

Beyond Literal Mappings: Metaphors, Sound-effects, Clichés, and Genre Sounds

In some cases, however, iconic mappings between sounds and computer events are impossible to achieve. For instance, what sound should copying a file make? It might be tempting to use the real-world analogy of photocopying a document. But this seems misleading in using the sound made by a repetitive, piece-meal operation to indicate a more continuous one (i.e., I can stop copying a paper document after one page, whereas electronic file copies are typically all-or-nothing). In addition, such a sound doesn't indicate some information that might be of value to users, such as the progress of the job. In this case, it seems useful to go beyond a literal mapping to create a sound that more fully indicates the attributes of copying in the model world of the computer.

One way to go beyond literal mappings is to use *metaphors*. In the SonicFinder, for instance, copying a file was indicated by the sound of liquid pouring into a container. As the operation neared completion, the sound rose in pitch to indicate that the container was getting full. Thus the sound conveyed useful attributes of the copy operation: that it involves transferring "stuff" (in this case data bits) into a new container, that the operation is relatively continuous, and that it is possible to predict how close it is to completion. However, metaphors such as these may also have unwanted implications: for instance, since the liquid must come from somewhere, the sound might be taken as indicating a move rather than a copy. The sound also does not differentiate reading and writing stages of a copy operation, which might be useful information in some circumstances. Finally, using the sound of a container being filled implies a constrained space waiting for the new information. In the case of the copy operation, this space is determined by the size of the information to be transferred, as if bottles changed size to automatically hold just the amount of liquid to be poured. As this example indicates, there is a trade-off in designing and using metaphors such as this: they provide a lawful structure that can guide users' expectations, but may sometimes be misleading.

Another approach to going beyond literal mappings is to use *sound-effects*. Although sound-effects are not naturally-occurring everyday sounds, if they are well designed they seem to be heard in many of the same ways. Think of the sound made by a ray-gun in a science fiction movie, for example. If well designed, it will indicate many of the properties of the event: that it is electronically produced, quick, powerful, hot enough to sear the air. Such sounds have never existed, but they are not arbitrary: they seem to extend the laws relating events to sounds.

Sound effects can be very useful in creating auditory icons. For example, in designing the SonicFinder it became clear that sounds should accompany opening and closing windows. Many attempts were made to use, literally, the sounds made by opening and closing (glass) windows, but none sounded appropriate. Finally it became clear that opening and closing windows in a computer is only metaphorically related to the real-world event. In the graphic interface, windows do not slide open, but instead expand into existence from their associated icon. They appear from nowhere in a way that has no literal analog in the real world – though such appearances are common in science fiction movies. Thus it was more appropriate to use a kind of "whooshing" noise to indicate this event, as if air rushed around the window as it expanded into existence.

Sound effects such as these are closely related to the *genre sounds* used by Cohen (xxx). As he points out, sounds from popular TV shows, movies and the like can be incorporated successfully into auditory interfaces. In some cases, they may provide information that is difficult to convey using more literal sounds. For instance, designing a sound for the teleporter in SoundShark might have a challenge because, clearly, no analogous mechanism exists in the everyday world – but it does in science fiction. Thus the sound used to convey this event, basically a slowly growing, inharmonic sound, was based on the sounds used to indicate teleportation in popular TV shows.

One of the drawbacks of genre sounds, as Cohen (xxx) notes, is that they may not be interpretable by people who are not familiar with their sources. This problem may be overcome to some degree by creating sound effects that are custom-made for particular applications. Insofar as they rely on extensions of physics, rather than familiarity with particular movies or television shows, such sounds can be expected to be more readily recognized even by users unfamiliar with the specific reference.

Finally, both sound-effects and genre sounds are closely related to another class of sounds that can be used to extend auditory icons. These are *auditory clichés*: sounds which are arbitrarily related to their meaning, even in the everyday world, but which are so firmly embedded in the culture that they can be used as everyday sounds are. For instance, from the point of view of everyday listening, the sound of a telephone bell simply conveys information about a hard mallet repetitively striking a small metal one. But because in many cultures this sound is strongly linked with the request to make an electronic connection to a remote colleague for the purposes of communication, it can be used to indicate analogous computer connections (as it is, for instance, in EAR). Similarly, from the point of view of everyday listening, the sound of somebody knocking on a door simply indicates a somewhat softer mallet striking a large wooden surface (for instance). Again, because the symbolic message it conveys is so widely recognized, auditory icons can be created which map to this message rather than the more literal meaning of the sound.

In sum, the vocabulary for auditory icons is driven more by the efficacy of the mappings allowed than by the sounds themselves. In general this means using everyday sounds that indicate their real-world meaning due to the laws of physics. But it is also possible and useful to expand the vocabulary for auditory icons to include metaphors, sound-effects, genre sounds and clichés. Such sounds can vastly expand the repertoire of auditory icons and the information they convey. Nonetheless, it must be recognized that the mapping from sound to meaning becomes more arbitrary the greater the move from everyday sounds and iconic mappings. In general, mapping a file selection to a tapping sound can be expected to be more readily guessed, learned, and remembered than mapping, for instance, an instantaneous move to the sound of a television transporter.

Annoyance

A final issue for auditory icons – and for auditory interfaces in general – is how to design sounds that will not prove annoying to users. As we pointed out in the Introduction, one of the first responses most people have to the idea of auditory interfaces is to say "I like to work in peace and quiet – why would I possibly want a computer that makes noise?"

One response to this is a variation of the story about John Cage that we told in the Introduction, whose experience in an anechoic chamber illustrated the fact that there's no such thing as silence. As a less extreme, but perhaps more telling model, listen carefully to the sounds around you right now. Even in a quiet office, you are likely to hear many subtle sounds: the whoosh of a ventilator, the hum of a computer, the shuffle of papers on nearby desks, footsteps as somebody walks by. Most of these sounds are hardly noticed, most of them are not annoying, and many of them are useful in maintaining a background awareness of ongoing events.

The point is not to say that concerns about auditory interfaces being annoying are misplaced. Rather it is to provide a model of how sounds can be used without being obtrusive: in general, the goal is to create sounds that are as subtle, unobtrusive, and informative as the sounds we already hear in the office. How can we do this? One answer is simply to use common sense in designing sounds: Don't make them too loud, too cute, too complex. Beyond this, however, we can turn to relevant basic research in understanding what makes sounds annoying.

Experimental studies of aesthetics, emotion, and psychoacoustics can provide a useful guide to the design of unobtrusive sounds (see Patterson, xxx; Berlyne, xxx; Mandler, xxx; Gaver & Mandler, xxx). Such studies break into two groups. The first concern the acoustic properties of sounds that seem to make them more or less annoying: this is the psychoacoustics of annoyance. The second group studies the interplay between acoustic variables of sounds, or patterns of sounds, and the experience of listeners in determining what will be annoying. This second group comes largely from those attempting experimental studies of aesthetics and emotion, though many psychoacousticians also recognize the relevance of such factors.

The Psychoacoustics of Annoying Sounds

Edworthy and her colleagues (1991) provide an excellent example of psychoacoustic studies relevant to annoyance. They were concerned with determining the acoustic correlates of *urgency* – the converse of annoyance – to guide the development of auditory warning systems such as those developed by Patterson (xxx; see Chapter 4). For each of their experiments, several acoustic factors were varied and subjects asked to rate the urgency of the resulting sounds. The factors they varied divided into two groups: the first affected individual sounds, or pulses, while the second applied to melodies, or bursts (see Table 6.1). Their findings showed that subjects were very consistent in judging urgency, which implies both that it is a psychologically real construct and that acoustic factors have consistent effects on it.

Table 6.1: Acoustic parameters affecting urgency (from Edworthy et al., 1990)

Parameter	**Most Urgent**		**Least Urgent**
Individual tones			
Fundamental Freq.	530Hz	–	150 Hz
Amp. Envelope	~rectangular	slow onset	slow offset
Harmonic series	random	10% irregular > %50 irregular	harmonic
Delayed harmonics	no delay	–	delayed harmonics
Sequences			
Speed	fast	moderate	slow
Rhythm	regular		syncopated
Number of units	4	2	1
Speed change	speeding	regular	slowing
Pitch range	large	small	moderate
Pitch contour	random		down/up
Musical structure	atonal	unresolved	resolved

The effects of acoustic parameters affecting urgency are summarized in Table 6.1. Each of these factors was found to be significant and consistent in the studies that Edworthy et al. (1990) ran. It should be noted, however, that the relative effects of the factors was not tested. Thus it is not clear, for instance, whether changes of pitch make larger differences in urgency than the inharmonicity of sounds. Nonetheless, these results provide extremely useful information for creating sounds and sequences that are more or less urgent (and thus annoying).

The Principle of Optimal Complexity

Results of psychoacoustic studies of urgency and annoyance can be extremely helpful in developing guidelines for the creation of auditory interfaces. However experiential factors also affect the annoyance of sounds and thus must also be taken into account. The old saying that "familiarity breeds contempt" has a lot of truth to it: We often find that the melody that was so instantly hummable becomes sickening after repeated hearing. Clearly experiential affects play a large role in determining what we will find annoying.

Experimental and theoretical studies of emotion and aesthetics are useful in understanding the role of experience. Two simple factors have emerged as most relevant: familiarity and complexity. In general, both are related to liking by an inverse-U curve: Both highly familiar and unfamiliar things are liked less than some moderately familiar item; both highly complex and simple things are less preferable than intermediate values. These findings can be summarized succinctly in the *principle of optimal psychological complexity:*

> *Moderately complex sounds are less annoying than very simple or very complicated ones; however, perceived complexity decreases with familiarity.*

This principle implies a variety of possibilities for the design of informative sounds. Complicated sounds may be used if they are heard relatively frequently: Although initially likely to be annoying, they will usually become acceptable over time. Conversely, very simple sounds should not be repeated too often: though initially acceptable, they become irritating if heard to often.

Perhaps the middle route is best: Use moderately complex sounds, but make subtle variations fairly often. This approach works well with the strategy behind parameterized auditory icons, in that it encourages more information to be conveyed by sounds, thus producing richer auditory icons. For example, in the SonicFinder the size of objects usually affected the sounds they made, with large objects making lower sounds than small ones. This had the effect not only of conveying potentially valuable information, but also of preventing the (often simple) impact and scraping sounds from becoming too annoying. It is fortunate that the least annoying auditory interfaces may also be the most informative.

Semantic Effects

To acoustic and experiential factors that influence annoyance must be added semantic factors. For instance, people regularly find highway sounds more annoying than birdsong, despite the fact that birdsongs have more of the acoustic parameters corresponding to annoyance (e.g., high pitch, abrupt envelopes, fast changes, etc.). It appears that the semantic connotations of the sounds can override acoustic and even experiential considerations in determining annoyance.

Note that semantic effects like these cannot be addressed from traditional views of sound and hearing because the meaning of sounds is outside the scope of such enquiries. Accounts of everyday listening (Chapter 5) can provide a somewhat better basis for understanding semantic effects on annoyance since they focus on the sources of sounds. But even everyday listening does not directly address higher-level semantics of many sounds – as can be seen, for instance, by the fact that a literal account of everyday listening would describe a telephone bell in terms of a piece of metal being struck repeatedly by a hard clapper. Still less can the emotional reactions to the semantics of sounds be predicted from theory. Nonetheless, designers can – and must – be sensitive to issues surrounding the semantics of sound in developing auditory interfaces.

The Tension Between Clarity and Obtrusiveness

The principles guiding the design of intuitively-obvious and unobtrusive auditory icons may often be at odds with one another. Everyday sounds tend to have many of the acoustic features that characterize annoying sounds. For instance, consider a metallic impact sound – used often in the auditory interfaces described above. For many object mappings, the sound will be high-pitched, inharmonic, abrupt, and atonal. This is the recipe for an urgent sound, according to our discussion above. On the other hand, consider trying to create unobtrusive sounds based on the acoustic guidelines we have described. According to these principles, the sounds should be low-pitched, harmonic, have smooth envelopes, and have tonal relations with one another. This implies severe constraints for the creation of auditory icons based on auditory icons.

There are several heuristics for reducing this tension between identifiability and obtrusiveness. First, everyday sounds may be shaped acoustically to reduce their annoyance. For instance, the attack of percussive sounds can be slowed slightly to reduce the tendency for abrupt sounds to demand attention. Most sounds can be low-pass filtered, reducing the amplitude of annoying high-frequency partials. Many long sounds can be shortened considerably, and thus made to convey information concisely. In all these cases, effective variations can be made without reducing sounds' identification if the changes are kept small.

In addition, the tension between the acoustic factors leading to identifiability and those that produce annoying sounds can be mitigated by taking advantage of experiential and semantic factors. This is, in fact, one of the major advantages of auditory icons: They are designed to be similar to the sorts of sounds one hears in the everyday world, and thus less distracting than introducing a new vocabulary (e.g. music). Because they fit with the existing auditory ambience, they benefit from familiarity and semantic congruity. Moreover, they can be designed to reach an optimal level of familiarity, as described above, and with the aim of using semantically pleasing – or better, neutral – sounds.

Conclusions

Auditory icons have great potential as a strategy for creating informative, intuitively-accessible, unobtrusive auditory interfaces. Already they have been shown to increase the tangibility of, and promote direct engagement with, the model world evoked by most graphical interfaces. They can provide feedback about users' actions, and feedforward about what might be done next. They serve as useful status monitors, helping people maintain a background awareness of ongoing processes and modes. They have been found to support a sense of the coherence of related processes. Finally, they are useful in providing a new medium for supporting collaboration; allowing colleagues to maintain awareness of one another while working independently.

The strategy behind auditory icons highlights a number of issues that have largely been left implicit by other auditory interfaces. In particular, the mapping between sounds and the information they convey is a crucial issue, with auditory icons suggesting the possibility and desirability of using iconic and metaphorical mappings rather than the symbolic mappings typically employed by the interfaces discussed in Chapters 3 and 4. In addition, they serve to indicate the range of auditory vocabularies that might be used, with abstract musical tones and literal recordings of everyday sounds being extremes in a continuum that also includes genre sounds, sound metaphors, and sound-effects. Finally, they serve to emphasize the acoustic, experiential, and semantic factors that help determine the obtrusiveness, and ultimately the acceptability, of auditory interfaces.

What's Next?

Despite the promise of the systems we have described in this chapter, however, there is much potential for the development of systems that use auditory icons. An excellent example of work in progress comes from the Mercateur group at Georgia Tech (Mynatt, ref). This group is engaged in building an auditory interface for blind users which will translate the information and style of presentation used in current graphical interfaces to the auditory domain. Their work promises to incorporate auditory icons, filtears (DO WE DESCRIBE THESE ANYWHERE?), and spatialized sound to create what might be thought of as a virtual reality for the blind – an artificial environment in which people may more and interact, but without the graphical components.

The development of auditory icons as a widespread interface technique, though, depends on making their creation more readily available to nonspecialized designers. There have been several developments to this end. First, as is commonly noted, computer power and memory is becoming greater and cheaper; in addition, it is becoming common for most hardware to include the capability to record and play sounds under software control. Second, the increasing power of today's computers is enabling the creation of synthesis algorithms specialized for generating parameterized everyday sounds for use as auditory icons (Gaver, 1993). Finally, the development of sound servers, analogous to window servers, will make the use of these algorithms and the deployment of sounds in general much easier to handle (Michel ref).

In the end, though, it is not just technical issues that will determine the success of auditory icons. While many researchers working with musical sounds call for research teams to include musicians for the design of auditory cues, we might be better off with good sound-effect designers. For the creation of systems that use auditory icons will depend on designers who have a good feel for how people hear sounds in the everyday world, the kinds of information that sounds can convey, the uses to which we put them, and the aesthetic possibilities of everyday sounds. With the help of such designers, we may expect to see more and more powerful systems that don't blare out incongruous, annoying sounds, but instead sounds with the subtlety and communicative power of the those we hear in the everyday world.