CHAPTER 9

Images, Words, and Gestures

This chapter addresses the relationship between visual information and verbal or textual information. Most visualizations are not purely graphical; they are composites, combining images with text or spoken language. But why do we need words? And when will images and words each be most effective? How should labels be used in diagrams? How should visual and verbal material be integrated in multimedia presentations? A particularly thorny but interesting problem is whether or not we should be using visual languages to program computers. Although computers are rapidly becoming common in every household, very few householders are programmers. It has been suggested that visual programming languages may make it easier for "nonprogrammers" to program computers.

We begin by considering the differences between visual and verbal means of communication, then move on to the application areas.

Coding Words and Images

Bertin, in his seminal work, *Semiology of Graphics* (1983), distinguishes two distinct sign systems. One cluster of sign systems is associated with auditory information processing and includes mathematical symbols, natural language, and music. The second cluster is based on visual information processing and includes graphics, together with abstract and figurative imagery. More recently, the dual coding of Paivio (1987) proposes that there are fundamentally different types of information stored in working memory; he calls them *imagens* and *logogens*. Roughly speaking, imagens denote the mental representation of visual information, whereas logogens denote the mental representation of language information.

Visual imagens consist of objects, natural groupings of objects, and whole parts of objects (for example, an arm), together with spatial information about the way they are laid out in a particular environment, such as a room. Logogens store basic information pertaining to language, although not the sounds of the words. Logogens are processed by a set of functional subsystems that provide support for reading and writing, understanding and producing speech, and logical thought. Logogens need not necessarily be tied to speech. Even in the profoundly deaf, the same language subsystems exist and are used in the reading and production of Braille and sign language.

The architecture of dual coding theory is sketched in Figure 9.1. Visual–spatial information enters through the visual system and is fed into association structures in the nonverbal imagen system. Visual text is processed, but is then fed into the association structures of logogens. Acoustic verbal stimuli are processed primarily through the auditory system and then fed into the logogen system. Logogens and imagens, although based on separate subsystems, can be strongly interlinked. For example, the word *cat* and language-based concepts related to cats will be linked to visual information related to the appearance of cats and their environment.

Much of this theory is uncontroversial. It has been known for decades that there are different neural processing centers for verbal information (speech areas of the temporal cortex) and visual information (the visual cortex). But the idea that we can "think" visually is relatively recent. One line of evidence comes from mental imaging. When people are asked to compare the size of a light bulb with the size of a tennis ball, or the green of a pea with the green of a Christmas tree, most claim that they use mental images of these objects to carry out the task (Kosslyn, 1994). Other studies by Kosslyn and his coworkers show that people treat objects in mental images as if they have real sizes and locations in space. Recently, positron emission tomography (PET) has been used to reveal which parts of the brain are active during specific tasks. This shows that when people are asked to perform tasks involving mental imaging, the visual processing

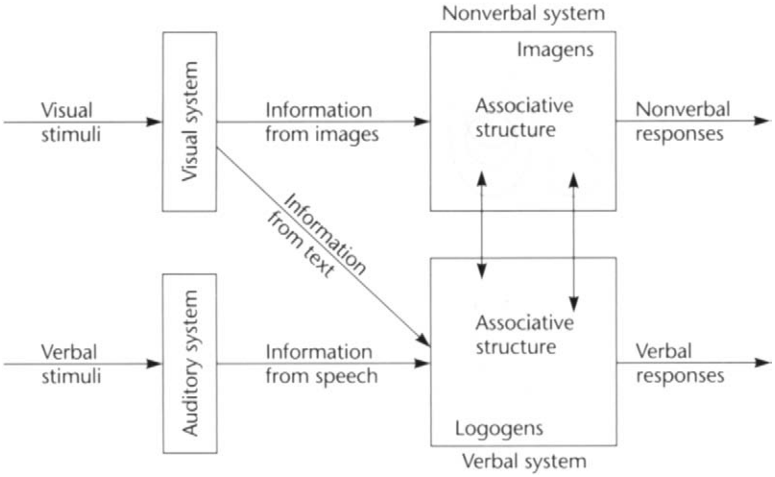

Figure 9.1 According to dual-coding theory, visual and verbal information is stored in different systems with different characteristics. *Adapted from Paivio (1987).*

centers in the brain are activated. Also, when they mentally change the size and position of an imagined object, different visual areas of the brain are activated (Kosslyn et al., 1993). In addition, if visual processing centers in the brain are damaged, mental imaging ability is disrupted (Farah et al., 1992). It would seem that when we see a cow and when we mentally visualize a cow, the same neural pathways are excited, at least in part.

Indeed, modern visual memory theory takes the position that visual object processing and visual object recognition are part of the same process. To some extent, the visual memory traces of objects and scenes are stored as part of the processing mechanism; thus it is not necessary for an object to be fully processed for recognition to take place (Beardsley, 1997). This can account for the great superiority of recognition over recall. We can easily recognize that we have seen something before, but reproducing it in a drawing or with a verbal description is much harder.

The Nature of Language

Noam Chomsky revolutionized the study of natural language because he showed that there are aspects of the syntactic structure of language that generalize across cultures (Chomsky, 1965). A central theme of his work is the concept that there are "deep structures" of language, representing innate cognitive abilities based on inherited neural structures. In many ways, this work forms the basis of modern linguistics. The fact that Chomsky's analysis of language is also a cornerstone of the theory of computer languages lends support to the idea that natural languages and computer languages have the same cognitive basis.

There is a critical period for normal language development that extends to about age 10. However, language is most easily acquired in the interval from birth to age three or four. If we do not obtain fluency in *some* language in our early years, we will never become fluent in any language.

Sign Language

Being *verbal* is not a defining characteristic of natural language. Sign languages are interesting because they are exemplars of true visual languages. If we do not acquire sign languages early in life, we will never become very adept at using them. Groups of deaf children spontaneously develop rich sign languages that have the same deep structures and grammatical patterns as spoken language. These languages are as syntactically rich and expressive as spoken language (Goldin-Meadow and Mylander, 1998). There are many sign languages; British sign language is a radically different language from American sign language, and the sign language of France is similarly different from the sign language of francophone Québec (Armstrong et al., 1994). Sign languages grew out of the communities of deaf children and adults that were established in the 19th century, arising spontaneously from the interactions of deaf children with one another. Sign languages are so robust that they thrived despite efforts of well-meaning teachers to suppress them in favor of lip reading—a far more limiting channel of communication.

Although in spoken languages words do not resemble the things they reference (with a few rare exceptions), signs are based partly on similarity. For example, see the signs for a tree

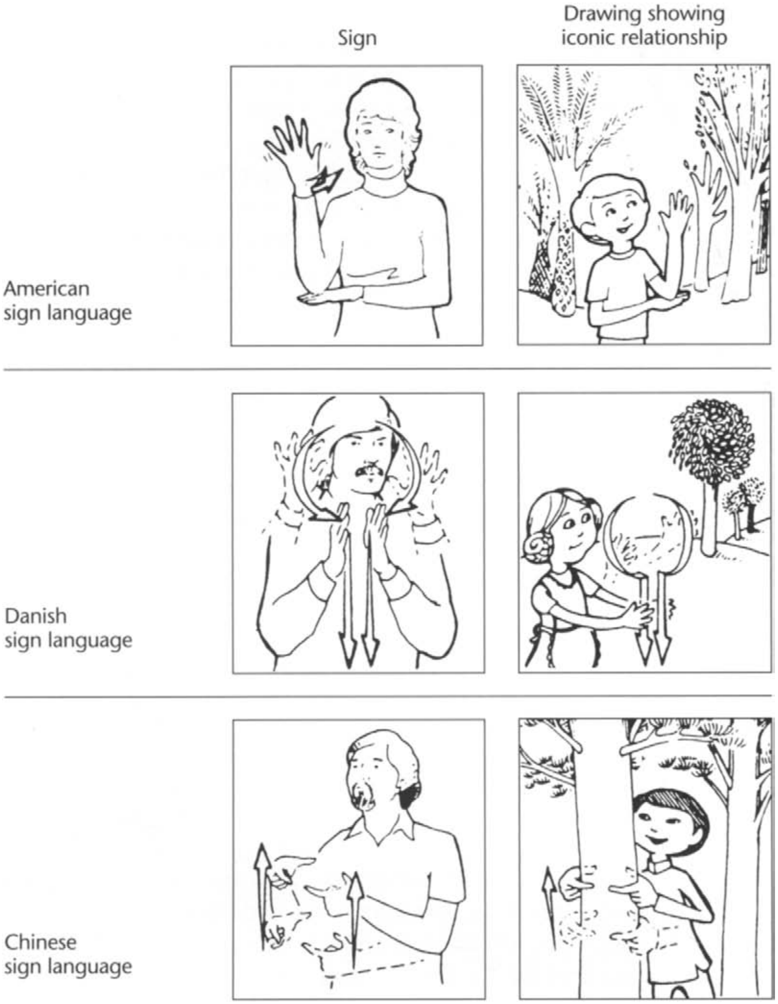

Figure 9.2 Three different sign-language representations of a tree. Note that they are all very different and all incorporate motion *From Bellugi and Klima (1976).*

illustrated in Figure 9.2. Sign languages have evolved rapidly. The pattern appears to be that a sign is originally created on the basis of a form of similarity in the shape and motion of the gesture, but over time, the sign becomes more abstract and similarity becomes less and less important (Deuchar, 1990). It is also the case that even signs apparently based on similarity are only recognized correctly about 10% of the time without instruction, and many signs are fully abstract.

Language Is Dynamic and Distributed over Time

We take in spoken, written, and sign language serially; it can take a few seconds to hear or read a short sentence. Armstrong et al. (1994) argue that in important ways, spoken language is essentially dynamic. Verbal expression does not consist of a set of fixed, discrete sounds; it is more accurately described as a set of vocal gestures producing dynamically changing sound patterns. The hand gestures of sign language are also dynamic, even when denoting static objects, as Figure 9.2 illustrates. There is a dynamic and inherently temporal phrasing at the syntactic level in the sequential structure of nouns and verbs. Even written language becomes a sequence of mentally recreated dynamic utterances when it is read.

In contrast with the dynamic, temporally ordered nature of language, relatively large sections of static pictures and diagrams can be understood in parallel. We can comprehend a complex visual structure in a fraction of a second, based on a single glance.

Visual and Spoken Language

The difficulty of writing and understanding computer programs has led to the development of a number of so-called *visual languages* in the hope that these can make the task easier. But we must be very careful in discussing these as languages. Visual programming languages are mostly static diagramming systems, so different from spoken languages that using the word *language* for both can be more misleading than helpful. Linguists and anthropologists commonly use the term *natural language* to refer to the spoken and written communications that make up our everyday human communication. Many of the cognitive operations required for computer programming have more in common with natural language than with visual processing.

Consider the following instructions that might be given to a mailroom clerk:

> Take a letter from the top of the In tray.
>
> Put a stamp on it.
>
> Put the letter in the Out tray.
>
> Continue until all the letters have stamps on them.

This is very like the following short program, which beginning programmers are often asked to write:

```
Repeat
  get a line of text from the input file
  change all the lowercase letters to uppercase
  write the line to the output file
Until (there is no more input)
```

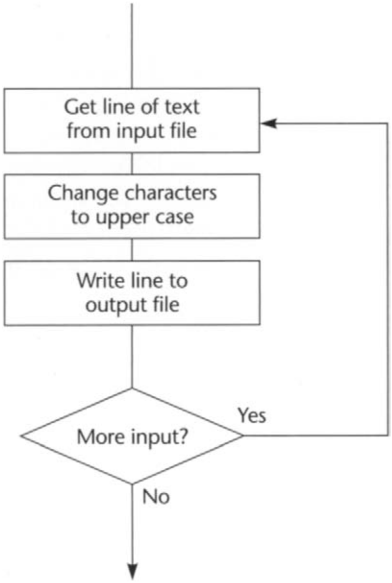
Get line of text from input file
Change characters to upper case
Write line to output file
More input?
Yes
No

Figure 9.3 A flowchart is often a poor way to represent information that can be readily expressed in natural language–like pseudocode.

This example program can also be expressed in the form of a graphical language called a *flowchart* (see Figure 9.3).

Flowcharts provide a salutary lesson to those who design visual programming languages. Flowcharts were once part of every introductory programming text, and it was often a contractual requirement that large bodies of software be documented with flowcharts describing the code structure. Once almost universally applied, flowcharts are now almost defunct. Why did flowcharts fail? It seems reasonable to attribute this to a lack of commonality with natural language. We have already learned to make *while* statements and *if–then* structured expressions in everyday communications. Using natural language–like pseudocode transfers this skill. But a graphical flowchart representing the same program must be translated before it can be interpreted in the natural-language processing centers.

Nevertheless, some information is much better described in the form of a diagram. A second example illustrates this. Suppose that we wish to express a set of propositions about the management hierarchy of a small company.

Jane is Jim's boss.

Jim is Joe's boss.

Anne works for Jane.

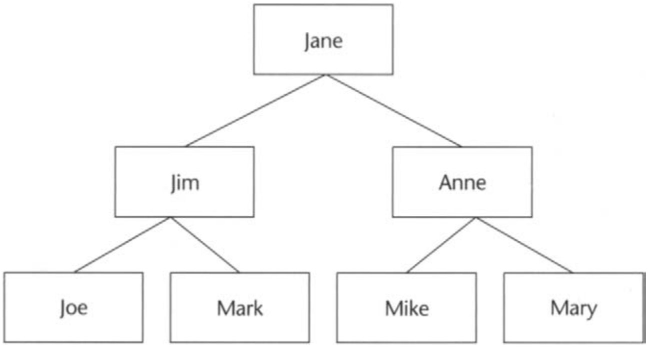

Figure 9.4 A structure diagram shows a hypothetical management hierarchy.

Mark works for Jim.

Anne is Mary's boss.

Anne is Mike's boss.

This pattern of relationships is far more clearly expressed in a diagram, as shown in Figure 9.4.

These two examples suggest that visual language, in the form of static diagrams, has certain expressive capabilities that are very different from, and perhaps complementary to, natural language. Diagrams should be used to express structural relationships among program elements, whereas words should be used to express detailed procedural logic.

However, the existence of the sign languages of the deaf suggests that there can be visual analogs to natural language and hence that effective visual programming languages are potentially possible. If they are to be developed, however, they must be dynamically phrased, rely heavily on animation, and ideally be learned early in life. We will return to this concept later in this chapter.

Images vs. Words

The greatest advantage of words over graphical communication, either static or dynamic, is that spoken and written natural language is ubiquitous. It is by far the most elaborate, complete, and widely shared system of symbols that we have available. For this reason alone, it is only when there is a clear advantage that visual techniques are preferred. In general, words should provide the general framework for the narrative of an extended communication. They can also be used for the detailed structure.

Having said that, often the visualization designer has the task of deciding whether to represent information visually, using words, or both. Other, related choices involve the selection of static or moving images and spoken or written text. If both words and images are used, methods for linking them must be selected. Useful reviews of cognitive studies that bear on these issues have been summarized and applied to multimedia design by a number of authors, including Strothotte and Strothotte (1997), Najjar (1998), and Faraday (1998). What follows is a summary of some of the key findings, beginning with the issue of when to use images vs. words. We start with static images, then consider animated images before moving to discuss the problem of combining images and words.

Static Images vs. Words

As a general comment, images are better for spatial structures, location, and detail, whereas words are better for representing procedural information, logical conditions, and abstract verbal concepts. Here are some more detailed points:

- Images are best for showing structural relationships, such as links between entities and groups of entities. Bartram (1980) showed that planning trips on bus routes was better achieved with a graphical representation than with tables.
- Tasks involving localization information are better conveyed using images. Haring and Fry (1979) showed improved recall of compositional information for pictorial, as opposed to verbal, information.
- Visual information is generally remembered better than verbal information, but not for abstract images. A study by Bower et al. (1975) suggested that it is important that visual information be meaningful and capable of incorporation into a cognitive framework for the visual advantage to be realized. This means that an image memory advantage cannot be relied on if the information is new and is represented abstractly and out of context.
- Images are best for providing detail and appearance. A study by Dwyer (1967) suggests that the amount of information shown in a picture should be related to the amount of time available to study it. A number of studies support the idea that first we comprehend the shape and overall structure of an object, then we comprehend the details (Price and Humphreys, 1989; Venturino and Gagnon, 1992). Because of this, simple line drawings may be most effective for quick exposures.
- Text is better than graphics for conveying abstract concepts, such as freedom or efficiency (Najjar, 1998).
- Procedural information is best provided using text or spoken language, or sometimes text integrated with images (Chandler and Sweller, 1991). Static images by themselves are not effective in providing complex, nonspatial instructions.
- Text is better than graphics for conveying program logic.
- Information that specifies conditions under which something should or should not be done is better provided using text or spoken language (Faraday, 1998).

Animated Images vs. Words

Computer animation opens up a whole range of new possibilities for conveying information. The work of researchers such as Michotte (1963), Heider and Simmel (1944), and Rimé et al. (1985), discussed in Chapter 6, shows that people can perceive events such as hitting, pushing, and aggression when geometric shapes are moved in simple ways. None of these things can be expressed with any directness using a static representation, although many of them can be well expressed using words. Thus, animation brings graphics closer to words in expressive capacity.

- Possibly the single greatest enhancement of a diagram that can be provided by animation is the ability to express causality (Michotte, 1963). With a static diagram, it is possible to use some device, such as an arrow, to denote a causal relationship between two entities. But the arrowhead is a conventional device that perceptually shows that there is *some* relationship, not that it has to do with causality. The work of Michotte shows that with appropriate animation and timing of events, a causal relationship will be directly and unequivocally perceived.
- An act of communication can be expressed by means of a symbol representing a message moving from the message source object to the message destination object (Stasko, 1990). For example, Figure 9.5 shows a part of a message-passing sequence between parts of a

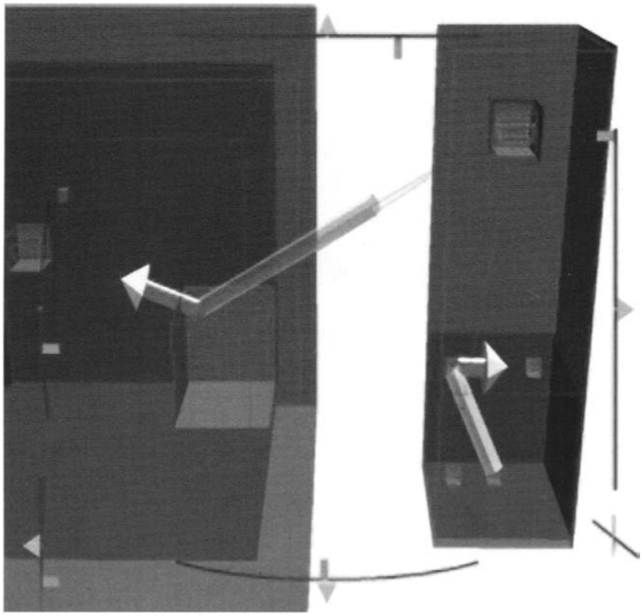

Figure 9.5 The "snakes" concept (Parker et al., 1998). *Image courtesy of NVision Software Systems.*

distributed program using a graphical technique called *snakes* (Parker et al., 1998). Animation moves the head of the snake from one software component to the next as the locus of computation moves; the tail of the snake provides a sense of recent history. Although a verbal or text description of this is possible, it would be difficult to describe adequately the behavior of *multiple* process threads, whereas multiple snakes readily can express this.

- A structure can be transformed gradually using animation. In this way, processes of restructuring or rearrangement can be made explicit. However, only quite simple mechanisms can be readily interpreted. Based on studies that required the inference of hidden motion, Kaiser et al. (1992) theorized that a kind of "naïve physics" is involved in perceiving action. This suggests that certain kinds of mechanical logic will be readily interpreted—for example, a simple hinge motion—but that complex interactions will not be interpreted correctly.
- A sequence of data movements can be captured with animation. The pioneering movie *Sorting Out Sorting* used animation to explain a number of different computer sorting algorithms by clearly showing the sequence in which elements were moved (Baecker, 1981). The smooth animated movement of elements enabled the direct comprehension of data movements in a way that could not be achieved using a static diagram.
- Some complex spatial actions can be conveyed using animation (Spangenberg, 1973). An animation illustrating the task of disassembling a machine gun was compared to a sequence of still shots. The animation was found to be superior for complex motions, but verbal instructions were just as effective for simple actions, such as grasping some component part. Based on a study of mechanical troubleshooting, Booher (1975) concluded that an animated description is the best way to convey perceptual-motor tasks, but that verbal instruction is useful to qualify the information. Teaching someone a golf swing would be better achieved with animation than with still images.

Links between Images and Words

The central claim of multimedia is that providing information in more than one medium of communication will lead to better understanding (Mousavi et al., 1995). Mayer et al. (1999) and others have translated this into a theory based on *dual coding*. They suggest that if active processing or related material takes place in both visual and verbal cognitive subsystems, learning will be better. It is claimed that dual coding of information will be more effective than single-modality coding. According to this theory, it is not sufficient for material to be simply presented and passively absorbed; it is critical that both visual and verbal representation be actively constructed, together with the connections between them.

Supporting multimedia theory, studies have shown that images and words in combination are often more effective than either in isolation (Faraday and Sutcliffe, 1997; Wadill and McDaniel, 1992). Faraday and Sutcliffe (1999) showed that multimedia documents with frequent and explicit links between text and images can lead to better comprehension. Fach and Strothotte (1994) theorized that using graphical connecting devices between text and imagery can explicitly form cross-links between visual and verbal associative memory structures. Care should be taken in linking words and images. For obvious reasons, it is important that words be associated with the appropriate images. These links between the two kinds of information can be static, as in the case of text and diagrams, or dynamic, as in the case of animations and spoken words.

Static Links

When text is integrated into a static diagram, the Gestalt principles discussed in Chapter 6 apply, as Figure 9.6 shows. Simple proximity is commonly used in labeling maps. A line drawn around the object and the text creates a common region; this can also be used to associate groups of objects with a particular label. Arrows and speech balloons linking text and graphics also apply the principle of connectedness.

Beyond merely attaching text labels to parts of diagrams, there is the possibility of integrating more complex procedural information. Chandler and Sweller (1991) showed that a set of instructional procedures for testing an electrical system were understood better if blocks of text were integrated with the diagram, as shown in Figure 9.7. In this way, process steps could be read immediately adjacent to the relevant visual information. Sweller et al. (1990) used the concept of *limited-capacity working memory* to explain these and similar results. They argue that when the information is integrated, there is a reduced need to store information temporarily while switching back and forth between locations.

There can be a two-way synergy between text and images. Faraday and Sutcliffe (1997) found that propositions given with a combination of imagery and speech were recalled better than propositions given only through images. Pictures can also enhance memory of text. Wadill and McDaniel (1992) provided images that were added redundantly to a text narrative; even though no new information was presented, the images enhanced recall.

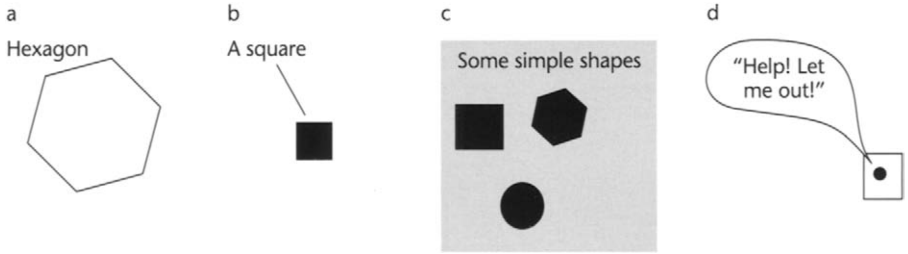

Figure 9.6 Various Gestalt principles are used to guide the linking of text and graphics: (a) Proximity. (b) Continuity/connectedness. (c) Common region. (d) Common region combined with connectedness.

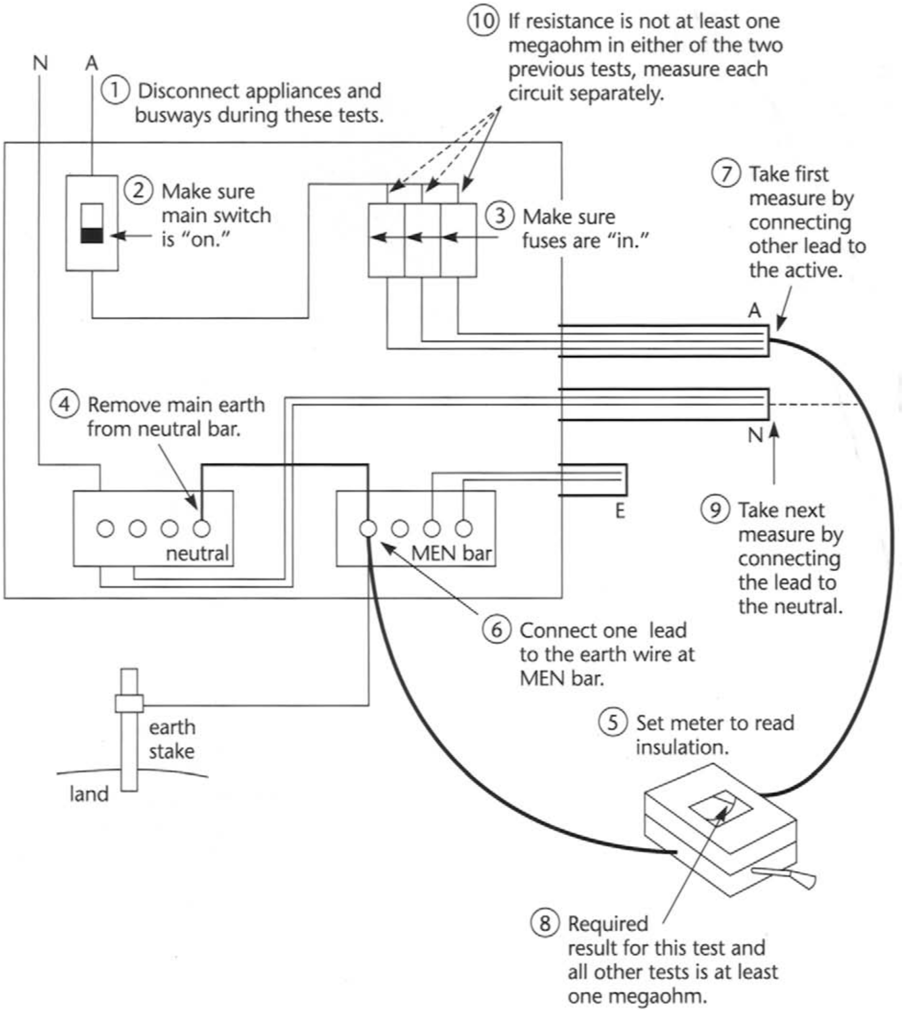
N
A
① Disconnect appliances and busways during these tests.
⑩ If resistance is not at least one megaohm in either of the two previous tests, measure each circuit separately.
② Make sure main switch is "on."
③ Make sure fuses are "in."
⑦ Take first measure by connecting other lead to the active.
A
④ Remove main earth from neutral bar.
N
E
⑨ Take next measure by connecting the lead to the neutral.
neutral
MEN bar
⑥ Connect one lead to the earth wire at MEN bar.
earth stake
land
⑤ Set meter to read insulation.
⑧ Required result for this test and all other tests is at least one megaohm.

Figure 9.7 An illustration used in a study by Chandler and Sweller (1991). A sequence of short paragraphs is integrated with the diagram to show how to conduct an electrical testing procedure.

The nature of text labels can strongly influence the way visual information is encoded. Jorg and Horman (1978) showed that when images were labeled, the choice of a general label (such as *fish*) or a specific label (such as *flounder*) influenced what would later be identified as previously seen. The broader-category label caused a greater variety of images to be identified (mostly erroneously). In some cases, it is desirable that people generalize specific instances into broader, more abstract categories, so this effect may sometimes be used to advantage.

Gestures as Linking Devices

When possible, spoken information—rather than text information—should accompany images, because the text necessarily takes visual attention away from the imagery. If the same information is given in spoken form, the auditory channel can be devoted to it, whereas the visual channel can be devoted to the imagery (Mousavi et al., 1995). The most natural way of linking spoken material with visual imagery is through hand gestures.

Deixis

In human communication theory, a gesture that links the subject of a spoken sentence with a visual reference is known as a *deictic gesture*, or simply *deixis*. When people engage in conversation, they sometimes indicate the subject or object in a sentence by pointing with a finger, glancing, or nodding in a particular direction. For example, a shopper might say "Give me that one," while pointing at a particular wedge of cheese at a delicatessen counter. The deictic gesture is considered to be the most elementary of linguistic acts. A child can point to something desirable, usually long before she can ask for it verbally, and even adults frequently point to things they wish to be given without uttering a word. Deixis has its own rich vocabulary. For example, an encircling gesture can indicate an entire group of objects or a region of space (Levelt et al., 1985; Oviatt et al., 1997).

To give a name to a visual object, we point and speak its name. Teachers will often talk through a diagram, making a series of linking deictic gestures. To explain a diagram of the respiratory system, a teacher might say, "*This tube* connecting the *larynx* to the *bronchial pathways* in the lungs is called the *trachea*," with a gesture toward each of the important parts.

Deictic techniques can be used to bridge the gap between visual imagery and spoken language. Some shared computer environments are designed to allow people at remote locations to work together while developing documents and drawings. Gutwin et al. (1996) observed that in these systems, voice communication and shared cursors are the critical components in maintaining dialog. It is generally thought to be much less important to transmit an image of the person speaking. Another major advantage of combining gesture with visual media is that this *multimodal* communication results in fewer misunderstandings (Oviatt, 1999; Oviatt et al., 1997), especially when English is not the speaker's native language.

Oviatt et al. (1997) showed that, given the opportunity, people like to point and talk at the same time when discussing maps. They studied the ordering of events in a multimodal interface to a mapping system, in which a user could both point deictically and speak while instructing another person in a planning task using a shared map. The instructor might say something like "Add a park here," or "Erase this line," while pointing to regions of the map. One of their findings was that pointing generally preceded speech; the instructor would point to something and then talk about it.

Interestingly, the reverse order of events may be appropriate when we are integrating text (as opposed to spoken language) with a diagram. In a study of eye movements, Faraday and Sutcliffe (1999) found that people would read a sentence, then look for the reference in an accompanying diagram. Based on this finding, they created a method for making it easy for users to make the appropriate connections. A button at the end of each sentence caused the relevant part of the image to be highlighted or animated in some way, thus enabling readers to switch attention rapidly to the correct part of the diagram. They showed that this did indeed result in greater understanding.

This research suggests two rules of thumb:

- If spoken words are to be integrated with visual information, the relevant part of the visualization should be highlighted just before the start of the relevant speech segment.
- If written text is to be integrated with visual information, links should be made at the end of each relevant sentence or phrase.

Deictic gestures can be more varied than simple pointing. For example, circular encompassing gestures can be used to indicate a whole group of objects, and different degrees of emphasis can be added by making a gesture more or less forceful.

Symbolic Gestures

In everyday life, we use a variety of gestures that have symbolic meaning. A raised hand signals that someone should stop moving. A wave of the hand signals farewell. Some symbolic gestures can be descriptive of actions. For example, we might rotate a hand to communicate to someone that they should turn an object. McNeill (1992) called these gestures *kinetographics*.

With input devices such as the Data Glove that capture the shape of a user's hand, it is possible to program a computer to interpret a user's hand gestures. This idea has been incorporated into a number of experimental computer interfaces. In a notable study carried out at MIT, researchers explored the powerful combination of hand gestures and speech commands (Thorisson et al., 1992). A person facing the computer screen first asked the system to

"Make a table"

This caused a table to appear on the floor in the computer visualization. The next command,

"On the table, place a vase,"

was combined with a gesture placing the fist of one hand on the palm of the other hand to show the relative location of the vase on the table. This caused a vase to appear on top of the table. Next, the command,

"Rotate it like this,"

was combined with a twisting motion of the hand causing the vase to rotate as described by the hand movement.

Although such systems are still experimental, there is evidence that combining words with gestures in this way will ultimately result in communication that is more effective and less error-prone (Mayer and Sims, 1994).

Expressive Gestures

Gestures can have an expressive dimension in addition to being deictic. Just as a line can be given a variety of qualities by being made thick, thin, jagged, or smooth, so can a gesture be made expressive (McNeill, 1992; Amaya et al., 1996). A particular kind of hand gesture, called a *beat*, sometimes accompanies speech, emphasizing critical elements in a narrative. Bull (1990) studied the way political orators use gestures to add emphasis. Vigorous gestures usually occurred at the same time as vocal stress. Also, the presence of both vigorous gestures and vocal stress often resulted in applause from the audience. In the domain of multimedia, animated pointers sometimes accompany a spoken narrative, but often quite mechanical movements are used to animate the pointer. Perhaps by making pointers more expressive, critical points might be brought out more effectively.

Visual Momentum in Animated Sequences

Moving the viewpoint in a visualization can function as a form of narrative control. Often a virtual camera is moved from one part of a data space to another, drawing attention to different features. In some complex 3D visualizations, a sequence of *shots* is spliced together to explain a complex process. Hochberg and Brooks (1978) developed the concept of *visual momentum* in trying to understand how cinematographers link different camera shots together. As a starting point, they argued that in normal perception, people do not take more than a few glances at a simple static scene; following this, the scene "goes dead" visually. In cinematography, the device of the cut enables the director to create a kind of heightened visual awareness, because a new perspective can be provided every second or so. The problem faced by the director is that of maintaining perceptual continuity. If a car travels out of one side of the frame in one scene, it should arrive in the next scene traveling in the same direction, otherwise the audience may lose track of it and pay attention to something else. Wickens (1992) has extended the visual momentum concept to create a set of four principles for user interface design:

1. Use consistent representations. This is like the continuity problem in movies, which involves making sure that clothing, makeup, and props are consistent from one cut to another. In visualization, this means that the same visual mappings of data must be preserved. This includes presenting similar views of a 3D object.

2. Use graceful transitions. Smooth animations between one scale view and another allow context to be maintained. Also, the technique of smoothly morphing a large object into a small object when it is "iconified" helps to maintain the object's identity.

3. Highlight anchors. Certain visual objects may act as visual reference points, or *anchors*, tying one view of a data space to the next. An anchor is a constant, invariant feature of a displayed world. Anchors become reference landmarks in subsequent views. When cuts are made from one view to another, ideally, several anchors should be visible from the previous frame. The concept of landmarks is discussed further in Chapter 10.
4. Display continuous overview maps. Common to many adventure video games and navigation systems used in aircraft or ground vehicles is the use of an *overview map* that places the user in a larger spatial context. This is usually supplemented by a more detailed local map. The same kind of technique can be used with large information spaces. The general problem of providing focus and context is also discussed further in Chapter 10.

Another technique used in cinematography is the *establishing shot.* Hochberg (1986) showed that identification of image detail was better when an establishing shot preceded a detail shot than when the reverse ordering was used. This suggests that an overview map should be provided first when an extended spatial environment is being presented.

Animated Visual Languages

When people discuss computer programs, they frequently anthropomorphize, describing software objects as if they were people sending messages to each other and reacting to those messages by performing certain tasks. This is especially true for programs written using object-oriented programming techniques. Some computer languages explicitly incorporate anthropomorphism. ToonTalk is one such language (Kahn, 1996). ToonTalk uses animated cartoon characters in a cartoon city as the programming model. Houses stand for the subroutines and procedures used in conventional programming. Birds are used as message carriers, taking information from one house to another. Active methods are instantiated by robots, and comparison tests are symbolized by weight scales. The developers of ToonTalk derived their motivation from the observation that even quite young children can learn to control the behavior of virtual robots in games such as Nintendo's Mario Brothers.

A ToonTalk example given by Kahn is programming the swapping of values stored in two locations. This is achieved by having an animated character take one object, put it to the side, take the second object and place it in the location of the first, and then take the first object and move it to the second location. Figure 9.8 illustrates this procedure.

KidSim is another interactive language, also intended to enable young children to acquire programming concepts using direct manipulation of graphical interfaces (Cypher and Smyth, 1995). Here is the authors' own description:

> *KidSim is an environment that allows children to create their own simulations. They create their own characters, and they create rules that specify how the characters are to*

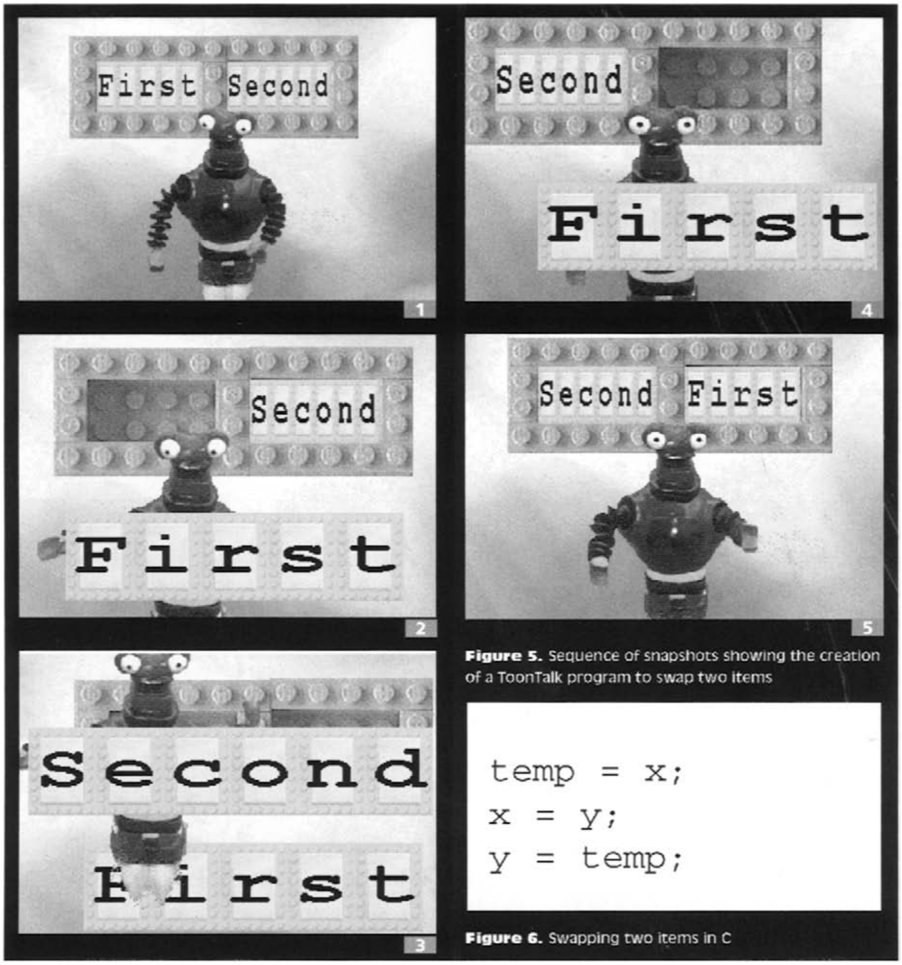

Figure 5. Sequence of snapshots showing the creation of a ToonTalk program to swap two items

```
temp = x;
x = y;
y = temp;
```

Figure 6. Swapping two items in C

Figure 9.8 A swap operation carried out in ToonTalk. In this language, animated characters can be instructed to move around and carry objects from place to place, just as they are in video games (Kahn, 1996b).

> *behave and interact. KidSim is programmed by demonstration, so that users do not need to learn a conventional programming language.*

In KidSim, as in ToonTalk, an important component is programming by example using direct manipulation techniques. In order to program a certain action, such as a movement of an object, the programmer moves the object using the mouse and the computer infers that this is a

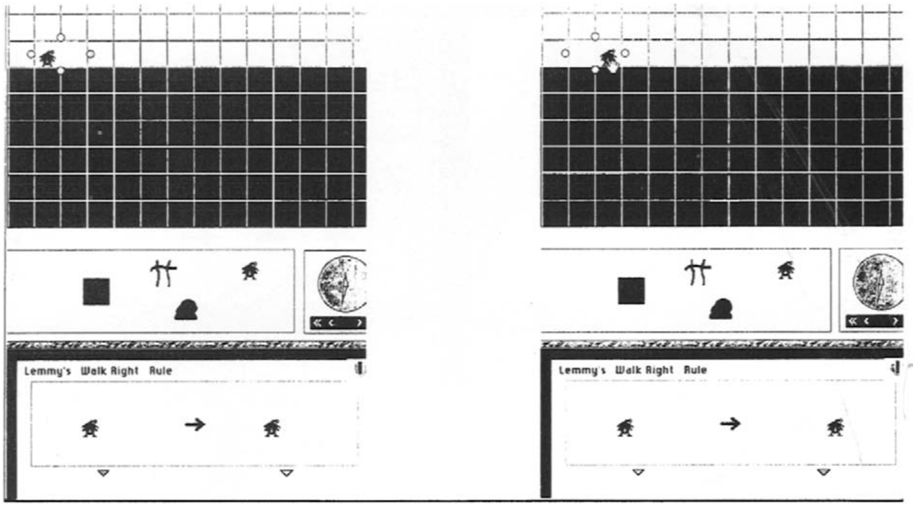

Figure 9.9 Creating a "Move Right" rule in KidSim. The user shapes a spotlight to outline the square to the right of the character, then drags the character into the adjacent square. At the bottom, the initial and final states for the rule are displayed (Cypher and Smyth, 1995).

programming event that should occur when a certain set of conditions is met. For example, when an actor gets close to a rock, the actor should jump over the rock.

Programming by example always requires that the programmer make a number of assumptions about how the system should behave. In KidSim, programs are based on graphical rewrite rules—a picture is replaced by another picture specified by demonstration. Figure 9.9 illustrates how the rule "If there is an empty space to the right of me, move me into it" is created. The programmer must first specify the area to which the rule applies and then drag the object from its old position to the new position. There are implicit assumptions that the user must make: the rule will apply wherever the picture object occurs on the screen, and the rule is repeated in an animation cycle.

The use of animated characters as program components can often lead to false assumptions about the programs that use them. Humans and animals get tired and bored, and can be expected to give up repetitive activities quite soon unless they are strongly motivated. Therefore, a child programming a computer with animated characters will expect them to stop and do something else after a while. But this is a poor metaphor for computers, which do not get tired or bored and can continue doing the same repetitive operation millions of times. Ultimately, both the strengths and the weaknesses of programming with animated characters will derive from the rich variety of visual metaphors that become available. Like all metaphors, they will be helpful if they are apt and harmful if they are not.

Rader et al. (1997) carried out an extensive independent evaluation of KidSim in two classrooms over the course of a year. The system was deliberately introduced without explicit teaching of the underlying programming concepts. They found that children rapidly learned the interactions needed to draw animated pictures but failed to gain a deep understanding of the programs. The children often tried to generalize the behavior they saw in ways that the machine did not understand. Students sometimes found it frustrating when they set up conditions they thought should cause some action, and then nothing happened.

A study by Palmiter et al. (1991) provided two kinds of instructions for a procedural task; one was an animated demonstration, the other was a written text. They found that immediately following instruction, the animated demonstration produced better performance. However, a week later, the results reversed; those who received written instructions did better. They explained these results by suggesting that in the short term, subjects could simply mimic what they had recently seen if they were given animated instructions. In the longer term, the effort of interpreting the written instructions produced a deeper symbolic coding of the information that was better retained over time.

Conclusion

Some of the advantages of visual representation, such as better comprehension of patterns and spatial relationships in general, seem clear and well documented. It is when we try to pin down the advantages of words that we run into difficulty. Indeed, some of the statements made, and supported by experimental results, appear to be contradictory. For example, some authors have suggested that procedural information is better described using words than images. But there appear to be counterexamples. The Gantt chart is a widely used graphical tool for project planning, and this is surely visual procedural information. Also, the study cited earlier by Bartram (1980), showing that visual representation of bus routes is better for planning a bus trip can be described as procedural planning.

It is possible that there is ultimately no kind of information for which words are demonstrably superior—all things being equal. But of course, they are not equal. Natural language provides us with the most developed and widely used symbol system available. We are all experts at it, having been trained intensively from an early age. We are not similarly experts at visual communication. The sign languages of the deaf show that a rich and complete visual equivalent is possible, but these alternative natural languages are inaccessible to most of us. Given the dominance of words as a medium of communication, visualizations will necessarily be hybrids, claiming ground only where a clear advantage can be obtained.

We should use images and words together whenever possible. Concepts presented using both kinds of coding are understood and remembered better. We evidently have cognitive subsystems dealing with both visual and verbal information (as discussed in Chapter 10), and it is possible that using both together may allow us to do more cognitive work. But to obtain a positive benefit from multimedia presentations, cross-references must be made so that the words and images can be integrated conceptually. Both time and space can be used to create these cross-links. The deictic gesture, wherein someone points at an object while speaking about it, is probably the most elementary of visual–verbal linking device. It is deeply embedded in human discourse and probably provides the cognitive foundation for other linking devices.

The material presented in this chapter suggests a number of conclusions about how we should design easy-to-learn computer programming languages. They should be hybrids of visual and natural language codes. Structure should be presented visually, and perhaps also created visually using direct manipulation techniques. Modules can be represented as visual objects, easily connected by drawing lines between them or by snapping them together. Detailed logical procedures should be programmed using words, not graphics. Ultimately, the use of speech recognition software may help beginning programmers with the difficulty of using a keyboard. They may use pointing gestures to bind the spoken words to the relevant parts of the diagrams.