Computers as Theatre

Brenda Laurel

Addison-Wesley Publishing Company
Reading, Massachusetts • Menlo Park, California • New York
Don Mills, Ontario • Wokingham, England • Amsterdam • Bonn
Sydney • Singapore • Tokyo • Madrid • San Juan • Milan • Paris

Chapter One

The Nature of the Beast

Representing Action

In 1962, the first computer game was invented by some hackers at MIT. It was called *Spacewar* and it ran on a DEC PDP-1, the world's first minicomputer, connected to a CRT display. One of the game's designers explained that the game was born when a group sat around trying to figure out "what would be interesting displays" they could create for the CRT with some pattern-generating software they had developed. "We decided that probably, you could make a two-dimensional maneuvering sort of thing, and decided that naturally the obvious thing to do was spaceships." The MIT hackers weren't the only ones to invent *Spacewar.* As Alan Kay noted, "the game of *Spacewar* blossoms spontaneously wherever there is a graphics display connected to a computer" [Brand, 1974].

Why was *Spacewar* the "natural" thing to build with this new technology? Why not a pie chart or an automated kaleidoscope or a desktop? Its designers identified *action* as the key ingredient and conceived *Spacewar* as a game that could provide a good balance between thinking and doing for its players. They regarded the computer as a machine naturally suited for representing things that you could see, control, and play with. Its interesting potential lay not in its ability to perform calculations but in its capacity to *represent action in which humans could participate.*

Why don't we look at everything computers do that way? Consider the following question:

Q: What is being represented by the Macintosh interface?

1. A desktop.
2. Something that's kind of like a desktop.
3. Someone doing something in an environment that's kind of like a desktop.

Number three is the only answer that comes close. The human is an indispensable ingredient of the representation, since it is only through a person's actions that all dimensions of the representation can be manifest. To put it another way, a computer-based representation without a human participant is like the sound of a tree falling in the proverbial uninhabited forest.

There are two major reasons for belaboring such a seemingly obvious point. First, it wasn't always true—and the design disciplines for applications and interfaces still bear the marks of that former time. Second, reconceptualizing what computers do as representing action with human participants suggests a design philosophy that diverges significantly from much of the contemporary thinking about interfaces.

Interface Evolution

"Interface" has become a trendy (and lucrative) concept over the last several years—a phenomenon that is largely attributable to the introduction of the Apple Macintosh. Interface design is concerned with making computer systems and applications easy to use (or at least usable) by humans. When we think of human-computer interfaces today, we are likely to visualize icons and menu bars, or perhaps command lines and blinking cursors. But it wasn't always so.

John Walker, founder and president of Autodesk, Inc., provides an illuminating account of the "generations" of user interface design [Walker, 1990]. In the beginning, says Walker, there was a one-on-one relationship between a person and a

computer through the knobs and dials on the front of massive early machines like the ENIAC. The advent of punch cards and batch processing replaced this direct human-computer interaction with a transaction mediated by a computer operator. Time-sharing and the use of "glass teletypes" reintroduced direct human-computer interaction and led to the command-line and menu-oriented interfaces with which the senior citizens of computing (people over thirty) are probably familiar. Walker attributes the notion of "conversationality" in human-computer interfaces to this kind of interaction, where a person does something and a computer responds—a tit-for-tat interaction.

This simplistic notion of conversation led many early interface specialists to develop a model of interaction that treats human and computer as two distinct parties whose "conversation" is mediated by the screen. But as advances in linguistics have demonstrated, there is more to conversation than tit-for-tat. Dialogue is not just linearized turn-taking in which I say something, you go think about it and then you say something, I go think about it, and so on. An alternative model of conversation employs the notion of *common ground*, described by Herbert H. Clark and Susan E. Brennan [1990]:

> It takes two people working together to play a duet, shake hands, play chess, waltz, teach, or make love. To succeed, the two of them have to coordinate both the content and process of what they are doing. Alan and Barbara, on the piano, must come to play the same Mozart duet. This is coordination of content. They must also synchronize their entrances and exits, coordinate how loud to play forte and pianissimo, and otherwise adjust to each other's tempo and dynamics. This is coordination of process. They cannot even begin to coordinate on content without assuming a vast amount of shared information or common ground—that is, mutual knowledge, mutual beliefs, and mutual assumptions [Clark and Carlson, 1982; Clark and Marshall, 1981; Lewis, 1969; Schelling, 1960]. And to coordinate on process, they need to update, or revise, their common ground moment by moment. All collective actions are built on common ground and its accumulation. [Clark and Brennan, 1990]

In her work in applying the notion of common ground to human-computer interfaces, Brennan [1990a] suggests that common ground is a jointly inhabited "space" where meaning takes shape through the collaboration and successive approximations of the participants. Brennan's ongoing work is aimed at designing human-computer interfaces so that they offer means for establishing common ground ("grounding") that are similar to those that people use in human-to-human conversation—for example, interruptions, questions, utterances, and gestures that indicate whether something is being understood [Brennan, 1990b].

Contemporary graphical interfaces, as exemplified by the Macintosh, explicitly represent part of what is in the "common ground" of interaction through the appearance and behavior of objects on the screen. Some of what goes on in the representation is exclusively attributable to either the person or the computer, and some of what happens is a fortuitous artifact of a collaboration in which the traits, goals, and behaviors of both are inseparably intertwined.

The notion of common ground not only provides a superior representation of the conversational process but also supports the idea that an interface is not simply the means whereby a person and a computer represent themselves to one another; rather it is a shared context for action in which both are agents. (This book will employ the noun "agent" to mean *one who initiates action*, a definition consistent with Aristotle's use of the concept in the *Poetics*. Insurance agents, real estate agents, and secret agents are examples of a kind of agency that is more complex—and vaguely ominous. The subject of "interface agents" is discussed later in Chapter 5.) When the old tit-for-tat paradigm intrudes, the "conversation" is likely to break down, once again relegating person and computer to opposite sides of a "mystic gulf" filled with hidden processes, arbitrary understandings and misunderstandings, and power relationships that are competitive rather than cooperative. "Mistakes," unanticipated outcomes, and error messages are typical evidence of such a breakdown in communication, where the common ground becomes a sea of misunderstanding.

The notion of interface metaphors was introduced to provide people with a conceptual scheme that would guard against such misunderstandings by deploying familiar objects and environments as stakes in the common ground. But even "good" metaphors don't always work. For instance, in an informal survey of Macintosh-literate university students, many people failed to employ the word "desktop" anywhere in their description of the Finder.[1] Where an interface metaphor diverges significantly from its real-world referent, people proceed by accounting for the behaviors of particular "objects" on the screen with ad hoc explanations of system operation, which are often incorrect—a "naive physics" of computing [see Owen 1986]. In such cases, metaphors do not serve as "stakes in the common ground," but rather as cognitive mediators whose labels may be somewhat less arcane (but possibly more ambiguous) than a computer scientist's jargon.

Although interface metaphors can fail in many ways (as discussed later in Chapter 5), their growing prevalence, especially in graphical interfaces, has expanded the domain of interface design to admit contributions from specialists in graphic and industrial design, linguistics, psychology, education, and other disciplines. An important contribution of the metaphorical approach has been to make interface design an *interdisciplinary* concern. The next section focuses on two of those "interdisciplines": psychology and graphic design.

Interface Interdisciplines

Psychology is a familiar domain to dramatists, actors, and other theatre artists because of its focus on human behavior. Understanding how psychology and theatre are alike and

[1]The Macintosh Finder is an application for managing people's file systems and for launching other applications. It comes with the system and is automatically launched when the machine is turned on. The Finder was designed on the basis of a "desktop metaphor," employing graphical icons to represent individual files as "documents" and hierarchical organizational units as "folders."

different may illuminate the distinct contributions that each can make in the field of human-computer interaction.

The two disciplines have several elements in common. Both concern themselves with how agents relate to one another in the process of communicating, solving problems, building things, having fun—the whole range of human activity. Both interpret human behavior in terms of goals, obstacles, conflicts, discoveries, changes of mind, successes, and failures. Both domains have important contributions to make to interface theory and design. Both attempt to observe and understand human behavior, but they employ that understanding to different ends: In general, psychology attempts to *describe what goes on in the real world* with all its fuzziness and loose ends, while theatre attempts to *represent something that might go on*, simplified for the purposes of logical and affective clarity. Psychology is devoted to the end of explaining human behavior, while drama attempts to represent it in a form that provides intellectual and emotional closure. Theatre is informed by psychology (both professional and amateur flavors), but it turns a trick that is outside of psychology's province through the art of representing action. By taking a look at some of the key ideas that psychology has contributed to interface design, we may be able to identify some ways in which theatrical knowledge can extend and complement them.

Psychologists have been involved in the quest to understand and shape human-computer interaction almost since the beginning of computing, through such disciplines as human factors and computer-aided instruction.[2] In the 1970s

[2]The literature on "human factors" and other psychological perspectives on human-computer interaction is huge. It is beyond the scope and purpose of this book to provide even a cursory survey of the entire domain. The work mentioned in this chapter is selected in terms of its relevance to the thesis of this particular book. Interested readers may wish to review *The Human Factor: Designing Computer Systems for People* by Richard Rubinstein and Harry Hersh [1984], which includes an excellent bibliography, *Readings in Human-Computer Interaction: A Multidisciplinary Approach* by Ronald M. Baecker and William A.S. Buxton [1987], or the various proceedings of ACM SIGCHI and the Human Factors Society.

and on through the 1980s, cognitive psychologists developed perspectives on human-computer interaction that were more critically focused on interface design than those of their colleagues in other branches of psychology. The work of Donald A. Norman, founder of the Institute for Cognitive Psychology at the University of California at San Diego, is especially illuminating. In the 1980s, Norman built a lab at UCSD that fostered some of the most innovative and germane thinking about human-computer interaction to date [see Norman and Draper, 1986, for a collection of essays by members and associates of this group]. Norman's perspective is highly task-oriented. In his book, *The Psychology of Everyday Things* [1988], Norman drives home the point that the design of an effective interface—whether for a computer or a doorknob—must begin with an analysis of what a person is trying to *do*, rather than with a metaphor or a notion of what the screen should display.

Norman's emphasis on action as the stuff that interfaces both *enable* and *represent* bores a tunnel out of the labyrinth of metaphor and brings us back out into the light, where *what is going on* is larger, more complex, and more fundamental than the way the human and the computer "talk" to each other about it.

Norman's insights dovetail nicely with those of the "common ground" linguists, suggesting a notion of the interface that is more than screen-deep. The interface becomes the arena for the performance of some task in which both human and computer have a role. What is represented in the interface is not only the task's environment and tools but also the process of interaction—the contributions made by both parties and evidence of the task's evolution. I believe that Norman's analysis supports the view that interface design should concern itself with representing *whole actions with multiple agents*. This is, by the way, precisely the definition of theatre.

Norman has also been a key figure in the development of another pivotal interface concept, the idea of *direct manipulation*. Direct manipulation interfaces employ a psychologist's knowledge of how people relate to objects in the real world in the belief that people can carry that knowledge across to the manipulation of virtual[3] objects that represent computational entities and processes.

The term *direct manipulation* was coined by Ben Shneiderman of the University of Maryland, who listed three key criteria:

1. Continuous representation of the object of interest.
2. Physical actions or labeled button presses instead of complex syntax.
3. Rapid incremental reversible operations whose impact on the object of interest is immediately visible [Shneiderman, 1987].

Shneiderman reports that direct-manipulation interfaces can "generate a glowing enthusiasm among users that is in marked contrast with the more common reaction of grudging acceptance or outright hostility" [Shneiderman, 1987]. In a cognitive analysis of how direct manipulation works, Hutchins, Hollan, and Norman [1986] suggest that direct manipulation as defined may provide only a partial explanation of such positive feelings. They posit a companion effect, labeled *direct engagement*, a *feeling* that occurs "when a user experiences direct interaction with the objects in a domain" [the notion of direct engagement is introduced in Laurel, 1986b]. Hutchins et al. add the requirements that input expressions be able to make use of previous output expressions, that the system create the illusion of instantaneous response (except where inappropriate to the domain), and that the interface be unobtrusive.

It seems likely that direct manipulation and direct engagement are head and tail of the same coin (or two handfuls of the same elephant)—one focusing on the qualities of action and the other focusing on subjective response. The basic issue

[3]The adjective *virtual* describes things—worlds, phenomena, etc.—that look and feel like reality but that lack the traditional physical substance. A virtual object, for instance, may be one that has no real-world equivalent, but the persuasiveness of its representation allows us to respond to it *as if* it were real.

is what is required to produce the feeling of taking action within a representational world, stripped of the "metacontext" of the interface as a discrete concern. Hutchins et al. sum it up this way: "Although we believe this feeling of direct engagement to be of critical importance, in fact, we know little about the actual requirements for producing it." Nevertheless, their analysis as well as Shneiderman's [1987] provide many valuable insights and useful examples of the phenomenon.

If we remove Shneiderman's clause regarding labeled button presses (because in many cases buttons are the artifacts of a pernicious interface metacontext), then the sense of directness can be boiled down to continuous representation, "physical" action, and apparent instantaneity of response. Apparent instantaneity depends upon both processing speed and the elimination of representations of intermediate activities in design. In the analyses of both Shneiderman and Hutchins et al., continuous representation and physical action depend heavily upon graphical representation. In fact, Hutchins et al. identify the granddaddy of direct manipulation as Ivan Sutherland's graphical design program, *Sketchpad* [Sutherland, 1963]. Graphical (and, by extension, multisensory) representations are fundamental to both the physical and emotional aspects of directness in interaction. Hence, it is worthwhile to examine the role and contributions of graphic design in the interface domain.

In many ways, the role of the graphic designer in human-computer interaction is parallel to the role of the theatrical scene designer. Both create representations of objects and environments that provide a context for action. In the case of theatre, the scene designer provides objects like teacups and chairs ("props"), canvas-covered wooden frames that are painted to look like walls ("flats"), and decorative things like draperies and rugs ("set dressing"). The behaviors of these elements is also designed—doors open, make-believe bombs explode, trick chairs break in barroom brawls. The lighting designer uses elements like color, intensity, and direction to illuminate the action and its environment and to focus our attention on key areas and events.

Both scene and light designers use such elements as line, shadow, color, texture, and style to suggest such contextual information as place, historical period, time of day, season, mood, and atmosphere. Theatrical designers also employ metaphor (and amplify the metaphors provided by the playwright) in the design of both realistic and nonrealistic pieces: the looming cityscape around Willy Loman's house in *Death of a Salesman* metaphorically represents his isolation and the death of his dreams; abstract webs of gauzy fabric suggest the multiple layers of illusion in the personality of Peer Gynt.

Likewise, in the world of interfaces, the graphic designer renders the objects and environments in which the action of the application or system will occur, imparting behaviors to some objects (like zoom-boxes and pop-up menus) and representing both concrete and ephemeral aspects of context through the use of such elements as line, shadow, color, intensity, texture, and style. Such familiar metaphors as desktops and windows provide behavioral and contextual cues about the nature of the activity that they support.

Both theatrical design and graphical interface design are aimed at creating representations of *worlds that are like reality only different*. But a scene design is not a whole play—for that we also need representations of character and action. Likewise, the element of graphical design is only part of the whole representation that we call an interface.

Throw the Baggage Out

The previous section picks up some of the more promising threads in the evolving discipline of interface design. It also suggests that these elements alone may not be sufficient in defining the nature of human-computer interaction or in realizing it effectively, and it recommends theatre as an additional perspective. But it may not be productive for theatre people simply to join all the other cooks in the kitchen. I want to take the argument a step further and suggest that the concept of *interface* itself is a hopeless hash, and that we might do better to throw it out and begin afresh.

A Definitional Digression

My frustration with the notion of the interface is as old as my involvement with computers. Perhaps the best way to explain it is to take a short excursion through the history of my personal view. I became involved with computers as a way to support myself while I was a graduate student in theatre. I thought that my career was going to take me to the stage, either as an actor or as a director. But a life in the theatre promised little in terms of income, and when a friend of mine started a little company to create computer software in 1977, I jumped at the chance to bolster my survival potential with some technical skills.

I became a software designer and programmer, working primarily on interactive fairy tales and educational programs for children. The company was called CyberVision, and the machine was a lowly 1802 processor with a four-color, low-resolution display and an alphanumeric keypad. The CyberVision computer was cassette-loaded with 2K of RAM, and it had the capacity to synchronize taped audio with animation on the screen. My first "feature" was an interactive, animated version of *Goldilocks*. Later, I created the first lip-synching on a microcomputer for a game of *Hangman* in which the evil executioner delivered menacing lines in a Transylvanian accent (all this with only sixteen lip positions). I immediately became immersed in translating my knowledge of drama and theatre to the task at hand because the two media were so obviously alike.

When CyberVision folded to its competition (an upstart company called Atari), I asked my boss to help me think about what kind of job to look for next. He said, "Why don't you go work for a bank? They need people to help design automated teller machines." "I don't know anything about *that*," I cried. "Of course you do," he replied. "That's human factors." In response to my blank look, he elaborated: "That's making computer things easy for people to use."

What a concept!

I ended up going to work for Atari, not for a bank, but the notion of ease of use as a design criterion fit neatly and permanently into my developing intuitions about how theatrical expertise could inform the art of designing software. There's nothing between the audience and the stage but some good illusion. Clearly, I was on the right track. But I hadn't run into the other "i" word yet.

After a few years in the software branch of the Atari home computer division, I decided to take time out to sit down and think through what I had come to believe about computers and theatre. (I also needed to begin my dissertation, which I had decided would be on that subject.) Alan Kay gave me the opportunity to do so in his research lab at Atari. "Interface" was every other word in the conversations of the bright young MIT wizards that populated the lab. I dimly perceived that there must be more to it than ease of use, and so signed up for a weekly seminar that one of the psychologists on the staff was conducting on the subject.

Models of the Interface

The seminar began by looking at how the concept of interface was typically understood by people in the computer field. Figure 1.1 shows a schematic model of the interface. The shaded rectangle in the middle represents the interface, which encompasses what appears on the screen, hardware input/output devices, and their drivers.

Compelling as its simplicity might make it, this model was immediately dismissed by everyone in the group. In order for an interface to work, the person has to have some

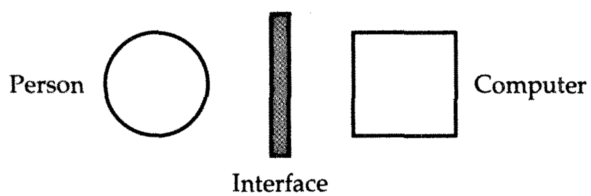

Figure 1.1 **The pre-cognitive–science view of the interface.**

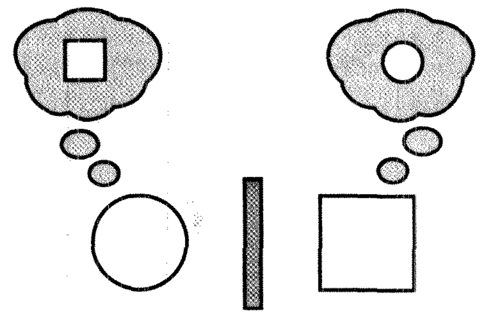

Figure 1.2 **The mental-models view.** The thought bubbles and their contents are considered part of the interface.

idea about what the computer expects and can handle, and the computer has to incorporate some information about what the person's goals and behaviors are likely to be. These two phenomena—a person's "mental model" of the computer and the computer's "understanding" of the person—are just as much a part of the interface as its physical and sensory manifestations (Figure 1.2). However, in order to use an interface

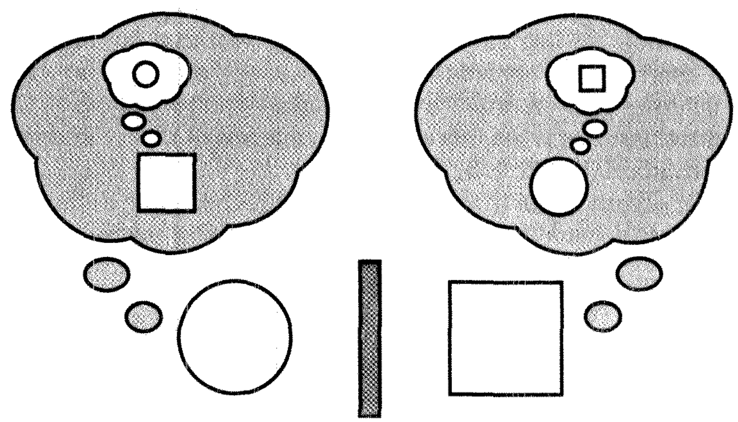

Figure 1.3 **The "horrible recursion" version of the mental-models view of the interface.** More bubbles could be added ad infinitum.

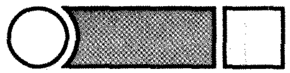

Figure 1.4 **A simple model of the interface, circa 1989.** In this view, the interface is that which joins human and computer, conforming to the needs of each.

correctly, you must also have an idea of what the computer is expecting you to do. If you are going to admit that what the two parties "think" about each other is part of what is going on, you will have to agree that what the two parties think about what the other is thinking about them must perforce be included in the model (Figure 1.3). This elaboration has dizzying ramifications.

Faced with this nightmare, our seminar at Atari abandoned the topic, and we turned our attention to more manageable concepts, such as the value of multisensory representations in the interface.

Over the years, I have frequently observed interface workers backing away from such gnarly theoretical discussions in favor of the investigation of more tractable issues of technique and technology—such subjects as direct manipulation, error handling, user testing, on-line help functions, graphics and animation, and sound and speech. The working definition of the interface has settled down to a relatively simple one—how humans and computers interact—but it avoids the central issue of what this all means in terms of reality and representation (Figure 1.4).

It occurs to me that when we have such trouble defining a concept, it usually means that we are barking up the wrong tree.

The World's a Stage

For purposes of comparison, let's take a look at the theatre. We have observed that the theatre bears some similarities to interface design in that both deal with the representation of

action. Drama, unlike novels or other forms of literature, incorporates the notion of *performance;* that is, plays are meant to be acted out. A parallel can be seen in interface design. In his book *The Elements of Friendly Software Design* [1982], Paul Heckel remarked, "When I design a product, I think of my program as giving a performance for its user." In the theatre, enactment typically occurs in a performance area called a stage (Figure 1.5). The stage is populated by one or more actors who portray characters. They perform actions in the physical context provided by the scene and light designers. The performance is typically viewed by a group of observers called an audience.

Part of the technical "magic" that supports the performance is embodied in the scenery and objects on the stage (windows that open and close; teacups that break); the rest happens in the backstage and "wing" areas (where scenery is supported, curtains are opened and closed, and sound effects are produced), the "loft" area above the stage, which accommodates lighting instruments and backdrops or set pieces that can be raised and lowered, and the lighting booth, which is usually above the audience at the back of the auditorium. The magic is created by both people and machines, but who, what, and where they are *do not matter* to the audience.

It's not just that the technical underpinnings of theatrical performance are unimportant to audience members; when a play is "working," audience members are simply not aware of the technical aspects at all. For the audience member who is

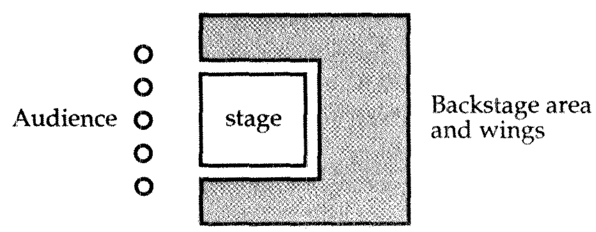

Figure 1.5 **Plan view of a typical proscenium theatre.**

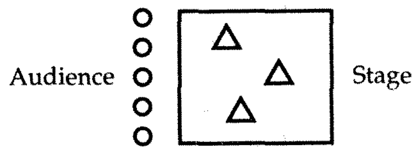

Figure 1.6 For the audience, what's happening on the stage is all there is. The triangles represent the actors.

engaged by and involved in the play, the action on the stage is *all there is* (Figure 1.6). In this sense, plays are like movies: When you are engrossed in one, you forget about the projector, and you may even lose awareness of your own body. For the actor on stage, the experience is similar in that everything extraneous to the ongoing action is tuned out, with the exception of the audience's audible and visible responses, which are often used by the actors to tweak their performance in real time (this, by the way, reminds us that theatrical audiences are not strictly "passive" and may be said to influence the action). For actor and audience alike, the ultimate "reality" is what is happening in the imaginary world on the stage—the representation.

As researchers grapple with the notion of interaction in the world of computing, they sometimes compare computer users to theatrical audiences. "Users," the argument goes, are like audience members who are able to have a greater influence on the unfolding action than simply the fine-tuning provided by conventional audience response. In fact, I used this analogy in my dissertation in an attempt to create a model for interactive fantasy. The users of such a system, I argued, are like audience members who can march up onto the stage and become various characters, altering the action by what they say and do in their roles.

Let's reconsider for a minute. What would it be really like if the audience marched up on the stage? They wouldn't know the script, for starters, and there would be a lot of awkward fumbling for context. Their clothes and skin would look

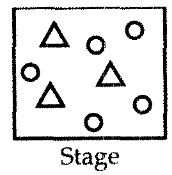

Figure 1.7 Putting the audience on the stage creates confusion.

funny under the lights. A state of panic would seize the actors as they attempted to improvise action that could incorporate the interlopers and still yield something that had any dramatic integrity. Or perhaps it would degenerate into a free-for-all, as performances of avant-garde interactive plays in the 1960s often did (Figure 1.7).

The problem with the audience-as-active-participant idea is that it adds to the clutter, both psychological and physical. The transformation needs to be subtractive rather than additive. People who are participating in the representation aren't audience members anymore. It's not that the audience joins the actors on the stage; it's that they *become* actors—and the notion of "passive" observers disappears.

In a theatrical view of human-computer activity, the stage is a virtual world. It is populated by agents, both human and computer-generated, and other elements of the representational context (windows, teacups, desktops, or what-have-you). The technical magic that supports the representation, as in the theatre, is behind the scenes. Whether the magic is created by hardware, software, or wetware is of no consequence; its only value is in what it produces on the "stage." In other words, *the representation is all there is* (Figure 1.8). Think of it as existential WYSIWYG.[4]

---

[4]WYSIWYG stands for "what you see is what you get," coined by Warren Teitelman at Xerox PARC. It has been held up as a paradigm for direct-manipulation interfaces, but some theorists have contested its value (see, for instance, Ted Nelson's article, "The Right Way to Think About Software Design" in *The Art of Human-Computer Interface Design*.

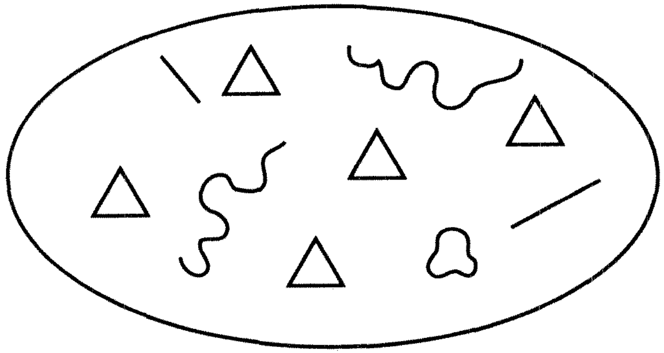

Figure 1.8 **An alternate view of human-computer interaction, in which the representation is all there is.** The triangles represent agents of either human or computer-generated types, and the other shapes are other objects in the virtual environment. The shape of the "stage" is oval, like the beam of a spotlight, to suggest that all that matters is that which is "illuminated."

Theatre as an Interface Metaphor

The idea of human-computer activity suggests a number of interesting corollaries. Since all action is confined to the world of the representation, all agents are situated in the same context, have access to the same objects, and speak the same language. Participants learn what language to speak by noticing what is understood; they learn what objects are and what they do by playing around with them. A good example of this approach is a system called *Programming by Rehearsal*, developed by Laura Gould and William Finzer at Xerox PARC in 1983 and 1984. The system is a visual programming environment based on a dramatic metaphor. There are some problems with the application of the metaphor per se,[5] but the principle

[5]Particularly troublesome is the idea of combining a group of performers and a "smart" stage into another performer. This is a case where a novel capability stretches the metaphor to its breaking point. This particular example effectively blurs the distinction between stage and performer and alerts us to the fact that the terms are being used "only" metaphorically.

of "the representation is all there is" is applied consistently with powerful results:

> Two significant obstacles to learning a programming language are mastering the language's syntax and learning the vocabulary. In the Rehearsal World, the designers rarely have to know either the syntax or the vocabulary as most writing of code is done by watching. [Finzer and Fitzer, 1984]

A more recent attempt to employ a theatrical metaphor for an authoring system is Ellis Horowitz's SCriptWriter system, developed at the University of Southern California in 1987 and 1988 [Horowitz, 1988]. Horowitz's system further illustrates the distinction between using theatre as an interface metaphor and using it in the deeper way that this book advocates—as a fundamental understanding of what is going on in human-computer interaction.

As a metaphor, Horowitz's system successfully employs notions like "director" (as the code of a program generated by his system) and "rehearsal" (in the same way that Gould's system employs the notion of programming by rehearsal). But Horowitz's interface falls off the edge of its own metaphor in several ways. Programming actions like "cast" and "rehearse" are intermixed with traditional computerese terms like "edit," "list," and "print," failing on the level of consistency. The most disturbing inconsistency is the notion of treating a screen as a "player." His player concatenates the notions of stage, scenery, actors, and dialogue in a concept where the locus of agency is so dispersed as to be invisible. Furthermore, the notion of *human* agency—the other kind of "player" that may act upon a "stage"—is absent in Horowitz's conceptualization. The system does not support a notion of action that integrates human agency into the whole but rather leaves this aspect of design entirely up to the author.

Interactivity and Human Action

The idea of enabling humans to take action in representational worlds is the powerful component of the programming-by-rehearsal approach. It is also what is missing in most attempts to use theatre simply as an interface metaphor. A central goal of this book is to suggest ways in which we can use a notion of theatre, not simply as a metaphor but as a way to conceptualize human-computer interaction itself.

Focusing on human agency allows us to simplify another consistently troublesome concept, the notion of "interactivity." People in the computer game business have been arguing about it for over a decade. In 1988, Alexander Associates sponsored INtertainment, the first annual conference bringing together people from all corners of the interactive entertainment business. People came from such diverse industries as personal computers, video games, broadcast and cable television, optical media, museums, and amusement parks. Over the course of the two days, a debate about the meaning of the word "interactive" raged through every session, disrupting carefully planned panels and presentations. People seemed to regard "interactivity" as the unique cultural discovery of the electronic age, and they demanded a coherent definition. Several speakers tried to oblige, but no one succeeded in presenting a definition that achieved general acceptance. Many participants departed angry and dissatisfied. Could it be the "wrong tree" problem again?

In the past, I've barked up that same tree. I posited that interactivity exists on a continuum that could be characterized by three variables: frequency (how often you could interact), range (how many choices were available), and significance (how much the choices really affected matters) [Laurel, 1986a and b]. A not-so-interactive computer game judged by these standards would let you do something only once in a while, would give you only a few things to choose from, and the things you could choose wouldn't make much difference to the whole action. A very interactive computer game (or desktop or flight simulator) would let you do something that really mattered at any time, and it could be anything you could think of—just like real life.

Now I believe that these variables provide only part of the picture. There is another, more rudimentary measure of interactivity: You either feel yourself to be participating in the

ongoing action of the representation or you don't. Successful orchestration of the variables of frequency, range, and significance can help to create this feeling, but it can also arise from other sources—for instance, sensory immersion and the tight coupling of kinesthetic input and visual response. If a representation of the surface of the moon lets you walk around and look at things, then it probably feels extremely interactive, whether your virtual excursion has any consequences or not. It enables you to *act within a representation* that is important. Optimizing frequency and range and significance in human choice-making will remain inadequate as long as we conceive of the human as sitting on the other side of some barrier, poking at the representation with a joystick or a mouse or a virtual hand. You can demonstrate Zeno's paradox on the user's side of the barrier until you're blue in the face, but it's only when you traverse it that things get real.[6]

The experience of interactivity is a thresholdy phenomenon, and it is also highly context-dependent. The search for a definition of interactivity diverts our attention from the real issue: How can people participate as agents within representational contexts? Actors know a lot about that, and so do children playing make-believe. Buried within us in our deepest playful instincts, and surrounding us in the cultural conventions of theatre, film, and narrative, are the most profound and intimate sources of knowledge about interactive representations. A central task is to bring those resources to the fore and to begin to use them in the design of interactive systems.

So now we have at least two reasons to consider theatre as a promising foundation for thinking about and designing human-computer experiences. First, there is significant overlap in the fundamental objective of the two domains—that is, representing action with multiple agents. Second, theatre suggests the basis for a model of human-computer activity that is familiar, comprehensible, and evocative. The rest of this book

[6]Zeno's paradox (called the theory of limits in mathematics) says that you can never get from here to there because you can only get halfway, then halfway of halfway, etc. Mathematics offers a solution; so does common sense. But the paradox is compelling enough to have interested logicians and mathematicians for centuries.

will explore some of the theoretical and practical aspects of theatre that can be directly applied to the task of designing human-computer experiences. But there are a few more stones to be turned in arranging the groundwork for this discussion.

Is Drama Serious Enough?

Because theatre is a form of entertainment, many people see it as fundamentally "non-serious." I have found that computer-science–oriented developers exhibit a high resistance to a theatrical approach to designing human-computer activity on the grounds that it somehow trivializes "serious" applications. Graphic designers undoubtedly have had to wrestle with the same sort of bias, design being seen not as a task of representation but one of mere decoration. Decoration is suspect because it may get in the way of the serious work to be done. (The same argument was used a few decades ago to ban bright colors, potted plants, and chatchkas from the workplace—but that's another story.) The fact of the matter is that graphics is an indispensable part of the representation itself, as amply demonstrated by the Macintosh and other contemporary computing environments.

The no-frills view that permeates thinking about the interfaces of "serious" applications is the result of a fundamental misunderstanding of the nature of seriousness in representations. The idea that theatre is "really not real" and is therefore unsuited as an approach to serious human-computer activities is misguided, because those activities are "really not real" in precisely the same ways. Without the representation, there is nothing at all—and theatre gives good representation.

Human-computer activity may be divided into two broad categories: productive and experiential [Laurel, 1986b]. Experiential activities, such as computer games, are undertaken purely for the experience afforded by the activity as you engage in it, while productive activities such as word processing have outcomes in the real world that are somehow beyond the experience of the activity itself. They are often

mistakenly defined in terms of their artifacts—a printed document or a spreadsheet filled with numbers. But seriousness is not equivalent to concreteness. A printed paper (such as a page in this book, for example) has "real" implications (for example, transmitting knowledge, changing how something is done, receiving a grade, or getting paid) even though it is *itself* a representation. "Productivity" as a class of applications is better characterized not by the concreteness of outcomes but by their seriousness vis-a-vis the real world.

There is a parallel here with seriousness as an aspect of drama. In formal terms, "serious" treatments of subjects are reserved for tragedy (and in some senses, melodrama) and "non-serious" treatments are found in melodrama, comedy, farce, and satire. Here again, although the plays themselves are representations, seriousness depends largely on the consequences of the actions represented in them. In a serious work—in *Hamlet* for instance—falling down (as Ophelia does after her father's death) has serious consequences both physically and symbolically; in a farce, falling down (tripping over a piece of furniture or slipping on a banana peel, for instance) causes no permanent injury or pain to the agent.

To trace these effects through to the real world, we need to look at their impact on audiences. Ophelia's fall and its symbolic meaning impart information about suffering, revenge, and the consequences of evil that can be contemplated, absorbed, and acted upon by an audience. The fall of a clown, on the other hand, may arouse laughter and ephemeral pleasure; it may also, as in more thoughtful flavors of comedy, communicate a philosophical view (for example, a lighthearted attitude toward random accidents). Seriousness in both theatre and human-computer activities is a function of the subject and its treatment in both formal and stylistic terms. Drama provides means for representing the whole spectrum of activity, from the ridiculous to the sublime.

Another objection to a theatrical approach is that theatre by its very nature is "fuzzy," while serious applications of computers require crystal clarity. The connotation of fuzziness probably derives from drama's emphasis on emotion—

subjective experience—while serious productivity is seen to require undiluted objectivity. Yet such "serious" tasks as formatting a paper for publication or designing a business plan for a new product can involve a far greater degree of subjectivity (in terms of creativity and evaluation, for instance) than "objective" skill and action (cutting and pasting, typing, and mousing around). At the farthest extreme, the notion that serious applications require objectivity, clarity, and precision is used as a rationale for rejecting natural-language interaction. This is because the success of machine understanding, at least in leading contemporary approaches, is probabilistic, whereas the understanding of symbolic logic (in mathematical or numerical representations) is seen to be unambiguous.

Yet people often drown in precision because of the complexity and artificiality of its expression (both lexical and syntactic). From the adventure-gamer grappling with a parser to the inexperienced UNIX user trying to "alias" a complicated e-mail address, people experience the requirement for precision as troublesome. This is no secret; the problem is commonly acknowledged and wrestled with by most interface designers [for example, see Rubinstein and Hersh, 1984, Chapter 6]. What may stop them from making a foray into the world of dramatic representation is the view that drama is fundamentally imprecise and therefore prone to error (both in terms of interpretation and subsequent action), while people require 100 percent success in all of their communications with computers. My experience suggests that, in the vast majority of contexts, this simply isn't true.

The imprecision of dramatic representation is the price people pay—often quite enthusiastically—in order to gain a kind of lifelikeness, including the possibility of surprise and delight. When "imprecision" works, it delivers a degree of success that is, in balance against the effort required to achieve it, an order of magnitude more rewarding than the precision of programming, at least for the nonprogrammer. When it doesn't work (as in the case of a parser error), how it is experienced depends heavily upon how the system handles the failure. "`I DON'T UNDERSTAND THAT WORD`" disrupts

and frustrates; an in-context response based on the most probable interpretation imitates a normal conversational failure and opens the way to methods of repair that are quite natural to use [see Brennan 1990b].

Both the frequency and robustness of the system's successes figure into the calculation of its value. A system that achieves only a moderate success rate (and no catastrophic failures) may be enthusiastically received if the successes are big ones, the effort required is minimal, and the overall experience is engaging. Chris Schmandt of the MIT Media Lab has developed a system that provides an extreme example. *Grunt* provides instructions for reaching a destination to the driver of a car. The system delivers directions via synthesized speech over a telephone. It listens to the driver for questions and cues about how well the driver has understood what it says. The trick is that the system is only listening to utterance pitch and duration and the duration of pauses—it *doesn't understand a single word*. Despite its low success rate (about 20 percent), *Grunt* has been received positively by most of its test users [see Schmandt, 1987 and 1990]. This is especially interesting in light of the fact that driving is viewed by most as a fairly serious activity, with strong real-world repercussions.

Seriousness in human-computer activities is a thresholdy thing. "Serious" and "non-serious" or "playful" activities can occur within the same context and at different stages in the same basic activity. I fool around with the layout of a document, for instance, experimenting with different fonts and paragraph styles, the placement of illustrations, perhaps even the structural divisions of the paper. At the point at which I make a creative decision and wish to implement a certain aspect of the design, I experience a "mode swing" (like a "mood swing," only different) toward greater "seriousness." I may then swing back to a "fooling around" mode as I evaluate the effects of a choice on the evolving document. In *Guides*, a research project at Apple investigating interfaces to multimedia databases, user testing revealed that people tend to move back and forth between browsing and focused searching "modes." They look around, then follow a line of investigation in an orderly and goal-directed fashion for a while, then begin to browse again. Apple researchers recorded similar behavior in the use of information kiosks that they installed at CHI '89 [Salomon, 1989]. Most of us have had similar experiences with encyclopedias, magazines, or even dictionaries.

The *Guides* project at Apple demonstrates this point. *Guides* was an investigation into the design and use of interface agents—computational characters that assist and interact with people. Most people who used the prototype *Guides* system were quite pleased to have the suggestions of the various guide characters about possible next moves in the database and to hear first-person stories that revealed a point of view about the content. They saw the guides as a great enhancement to the experience of browsing. At the same time, many saw the guides as impediments to goal-directed searching; that is, when they knew what they were looking for, the guides seemed to get in the way. There was simply too much indirection. One had to hope that the guide was "smart" enough to figure out what was sought and not to overlook anything relevant, and there was no way to find out for sure how well the guide was doing. There was also no way to chunk the guides' stories or to search through them with efficiency or acuity for smaller pieces of information. The *Guides* researchers were tempted to view these results as a need for a clearer, more "objective" approach to goal-directed searching.

In fact, that was only part of what people wanted. Many expressed the desire to be able to say to a guide, "go find this and only this and don't bother telling me how you feel about it," but there was no way to have such a conversation. The problem arose precisely at the threshold of "seriousness" that is crossed when shifting from an *experiential* mode (browsing) to an *instrumental* mode (goal-directed searching). There are two basic approaches to a solution. The first is to shift the locus of control from delegation to direct human agency. This approach would replace the guides with a representational environment containing the means for people to search the database themselves by topic or keyword. It corresponds

roughly to the concept of a library, where you can move around among shelves of books that are arranged topically, occasionally picking one up and leafing through it. Of course, the virtual library is augmented by representational magic—books open to a page indicated by spoken keywords, stairs need not be climbed, and there is always enough light.

The second approach is to create a kind of agent that is capable of understanding a person's goal and delivering a successful result smoothly and efficiently—a kind of augmented reference librarian. Such an agent could theoretically deliver a bigger win in a couple of ways. First, it could preserve continuity in the representational context by continuing to employ agents. Second, it could still shift the locus of control to the person by making the searching agent more explicitly subservient and responsive. Third, it creates the possibility for results beyond those that could be achieved by a person examining the database in the first person—no matter how magical the virtual library might be, searching it thoroughly would quickly exceed the human thresholds for both tedium and complexity. For instance, a searching agent might not only look at the topical index but also access topical data that might be associated with smaller chunks of information—data that would be too numerous for a person to examine in detail. A searching agent might then provide an array of possible information sources as the result of its search, each cued up to the most relevant chunk. Here the potential for surprise and delight is optimized, making the experience more pleasurable. Such powers of agents will be discussed in greater detail in the section on agents in Chapter 5.

In summary, a dramatic approach need not be fuzzy or imprecise in its ability to produce results. It is potentially capable of supporting both serious and nonserious activities. Its evocative powers and even its ambiguities can be harnessed to enhance rather than to impede a person's serious goals, and to create the possibility of surprise and delight—things that are rarely produced by exhaustive responses to crystal-clear specifications.

For many people whose way of working can be characterized as objective or scientific, the idea of employing an artistic approach is troublesome. It's hard to say how artists do what they do. The process seems to consist largely of imagination and inspiration, and there seems to be no forthright, dependable methodology. Yet, as we observed in the Foreword, and as we will expand upon in the next chapter, there are ways in which art is "lawful"; that is, there are formal, structural, and causal dimensions that can be identified and used both descriptively and productively. The final goal of this chapter is to justify taking an *artistic* approach to the problem of designing human-computer activity.

An Artistic Perspective

In his seminal book, *The Elements of Friendly Software Design* [1982], Paul Heckel characterizes software design as primarily concerned with communication. He observes that "among all the art forms that can teach us about communication, the most appropriate is filmmaking." Heckel chooses filmmaking as an example over older forms (such as theatre) because it "illustrates the transition from an engineering discipline to an art form." He goes on to observe that movies did not achieve wide popular success until artists replaced engineers as the primary creators. Heckel's book is filled with references to illusion, performance, and other theatrical and filmic metaphors with software examples to illustrate each observation. He gives the use of metaphor in interface design a different twist by employing filmmaking, writing, acting, and other "communication crafts" as metaphors for the process of software design.

In 1967, Ted Nelson examined the evolution of film in order to understand how the new medium he envisioned—hypertext—should develop. In considering the ways in which the stage had influenced film, he noted that "stage content, when adapted, was appropriate and useful, while stage techniques (such as the notion of a proscenium

and an insistence on continuous action within scenes) were not [Nelson in Schecter, 1967]. From the vantage point of 1990, we can see a migration of both techniques and content from film into the computer medium. If one takes the theatre and the film medium as subsets of a larger category, as representations of action in virtual worlds, then another key similarity between these media and computers is their fundamental elements of form and structure and their purpose.

Both Heckel and Nelson draw our attention to the centrality of "make-believe" in the conception and design of software. An engineer's view of software design is rooted in logic, realizing an orderly set of functions in an internally elegant program. In Heckel's view, the better approach is rooted in vision, which realizes an environment for action through evocative, consistent illusions. According to Nelson, it is the creation of "virtualities"—representations for things that may never have existed in the real world before [Nelson, 1990]. The role of imagination in creating interactive representations is clear and cannot be overrated. In an important sense, a piece of computer software is a collaborative exercise of the imaginations of the creator(s) of a program and people who use it.

Imagination supports a constellation of distinctively human phenomena that includes both symbolic thinking and representation-making. There is a story about a monkey and some bananas that every undergraduate psychology student has heard. A researcher places a monkey in a room with a bunch of bananas hanging from the ceiling and a box on the floor. The monkey tries various ways of getting the bananas—reaching, jumping, and so on—and eventually climbs up onto the box. A person in a similar situation would rehearse most of the possible strategies in her head and actively pursue only those that seemed promising, maybe only the successful one. For the monkey, the focus of attention is the real bananas; for the human, it's what's going on inside her head. Imagination is a shortcut through the process of trial and error.

But imagination is good for much more than real-world problem solving. The impulse to create interactive representations, as exemplified by human-computer activities, is only the most recent manifestation of the age-old desire to make what we imagine palpable—our insatiable need to exercise our intellect, judgment, and spirit in contexts, situations, and even personae that are different from those of our everyday lives. When a person considers how to climb a tree, imagination serves as a laboratory for virtual experiments in physics, biomechanics, and physiology. In matters of justice, art, or philosophy, imagination is the laboratory of the spirit.

What we do in our heads can be merely expedient or far-reaching, private or intended for sharing and communication. The novels of Ayn Rand, for instance or the plays of George Bernard Shaw create worlds where people address issues and problems, both concrete and abstract, and enact their discoveries, responses, and solutions. These representations are wholly contained in the realm of the imagination, yet they transport us to alternate possible perspectives and may influence us in ways that are more resonant and meaningful than experiences actually lived.

Art is the external *representation* of things that happen in the head of the artist. Art forms differ in terms of the materials they employ, the way the representations are created, what they purport to represent, and how they are manifest in the world. Different forms have different powers—the powers to engage, to provide pleasure and information, to evoke response. But all have as their end the *representation* of some internal vista that the artist wishes to create beyond the bounds of his or her own skull, making it available in some form to other people.

What are such representations good for? Aristotle defined *catharsis* as the end cause of a play and saw it as the pleasurable release of emotion, specifically those emotions evoked by the action represented in the play.[7] In his view, catharsis occurred during the actual "run-time" of the play, but some contemporary theorists disagree. The early twentieth-century

[7]That's not to say that plays must arouse only pleasant emotions; the pleasure of release makes even nasty emotions enjoyable in a theatrical context. Catharsis is discussed more fully in Chapter 4.

German dramatist Bertolt Brecht extended the notion of catharsis beyond the temporal boundary of the performance [Brecht, 1964]. He posited that catharsis is not complete until the audience members take what they have assimilated from the representation and put it to work in their lives. In Brecht's hypothesis, the representation lives between imagination and reality, serving as a conductor, amplifier, clarifier, and motivator.

It seems to me that computer-based representations work in fundamentally the same way: a person participates in a representation that is not the same as real life but which has real-world effects or consequences. Representation and reality stand in a particular and necessary relation to one another. In much contemporary thinking about interfaces, however, the understanding of that relationship is muddy. On the one hand, we speak of "tools" for "users" to employ in the accomplishment of various tasks with computers. We plumb psychology for information about how people go about using tools and what is the best way to design them. We arrive at notions like "cut" and "paste" and even "write" that seem to suggest that people working with computers are operating in the arena of the concrete. We often fail to see that these are *representations* of tools and activities and to notice how that makes them different from (and often better than) the real thing.

On the other hand, we employ graphic artists to create icons and windows, pictures of little hands and file folders and lassos and spilling paint cans, to stand in for us in the computer's world. Here the idea of representation is used, but only in a superficial sense. Messy notions like "interface metaphor" are employed to gloss over the differences between representation and reality, attempting to draw little cognitive lines from the things we see on the screen to the "real" activities that psychologists tell us we are performing. Interface metaphors rumble along like Rube Goldberg machines, patched and wired together every time they break, until they are so encrusted with the artifacts of repair that we can no longer interpret them or recognize their referents.

This confusion over the nature of human-computer activity can be alleviated by thinking about it in terms of theatre, where the special relationship between representation and reality is already comfortably established, not only in theoretical terms but also in the way that people design and experience theatrical works. Both domains employ representations as contexts for thought. Both attempt to amplify and orchestrate experience. Both have the capacity to represent actions and situations that do not and cannot exist in the real world, in ways that invite us to extend our minds, feelings, and senses to envelop them.

In the view of semioticist Julian Hilton [1991], theatre is "essentially the art of showing, the art of the index . . . . It involves the synthesis of symbolic and iconic systems (words and moving pictures) in a single indivisible performed event." Hilton employs the myth of Pygmalion and Galathea (familiar to many as the basis of George Bernard Shaw's *Pygmalion* and its musical adaptation, *My Fair Lady*) to express the relationship of the theatre to the domain of artificial intelligence. He describes the value of the theatre's ability to represent things that have no real-world referents in semiotic terms:

> Galathea in a literal sense imitates nothing, and as such defines a class of icon (the statue after all is a picture of itself) that can simultaneously be an index. It is this category of non-imitative index which enables the index to liberate its true power, whereby it has all the infinite valency of the symbol while retaining the immediate recognisability of the icon. [Hilton, 1991]

Computers are representation machines that can emulate any known medium, as Alan Kay observes:

> The protean nature of the computer is such that it can act like a machine or like a language to be shaped and exploited. It is a medium that can dynamically simulate the details of any other medium, including media that cannot exist physically. It is not a tool, although it can act like many tools. It is the first metamedium, and as such it has degrees of freedom for representation and expression never before encountered and as yet barely investigated. [Kay, 1984]

Thinking about interfaces is thinking too small. Designing human-computer experience isn't about building a better

desktop. It's about creating imaginary worlds that have a special relationship to reality—worlds in which we can extend, amplify, and enrich our own capacities to think, feel, and act. Hopefully, this chapter has persuaded you that knowledge from the theatrical domain can help us in that task. The next two chapters are designed to give you a deeper understanding of some of the most relevant aspects of dramatic theory and to apply them to interactive forms.

may implicitly constrain their thoughts, choices, and actions. In systems that employ simple language parsers, for instance, words that are unknown to the system cannot effect any change in the world; choices and actions that are represented by unknown words cannot be performed.

It is difficult to avoid such a disruptive effect when people are allowed or encouraged to make choices that they cannot effectively express to the system. For instance, the text adventure games developed by Infocom are presented entirely in a verbal mode. People are encouraged to use natural language to express their choices, and so they expect words to work. They have no clue to tell them which words are unknown to the system *except the experience of failure*. On the other hand, given the text-based nature of the game and the equipment that it is usually run on, people are never encouraged to attempt to express themselves through gestures or physical actions. The absence of visual and kinesthetic modes in the system is accepted as a given, and the resulting constraints are unobtrusive. Such constraints are extrinsic to the action but may be utilized effectively if they are presented simply and explicitly, or if they are integrated into the mimetic context (for example, "this ship is not equipped for voice communication").

Generally, the more modes that are present in the interface (verbal, visual, auditory, etc.), the more complex the system must be in order to handle the reception and interpretation of a wide variety of inputs and to formulate and orchestrate its responses. Constraining people through limitations on input and output capabilities becomes less effective as the number of modes in the interface increases; separate sets of constraints for each mode serve to confuse and frustrate people. In a multimodal interface environment, intrinsic formal and material constraints are therefore preferable to those based on the technical characteristics of the interface.

Engagement: The First-Person Imperative

In the foregoing discussion, *engagement* was held up as a desirable—even essential—human response to computer-mediated activities. Engagement has cognitive components,

but it is primarily understood as an emotion. Why should we demand that all human-computer activities elicit this particular emotional response? What is its nature, and what is its value? What can designers do to guarantee that it occurs?

Engagement, as I use the concept in this book, is similar in many ways to the theatrical notion of the "willing suspension of disbelief," a concept introduced by the early nineteenth-century critic and poet Samuel Taylor Coleridge.[6] It is the state of mind that we must attain in order to enjoy a representation of an action. Coleridge believed that any idiot could see that a play on a stage was not real life. (Plato would have disagreed with him, as do those in whom fear is induced by any new representational medium, but that is another story.) Coleridge noticed that, in order to enjoy a play, we must temporarily suspend (or attenuate) our knowledge that it is "pretend." We do this "willingly" in order to experience other emotional responses as a result of viewing the action. When the heroine is threatened, we feel a kind of fear for and with her that is recognizable as fear but different from the fear we would feel if we were tied to the railroad tracks ourselves. *Pretending that the action is real* affords us the thrill of fear; *knowing that the action is pretend* saves us from the pain of fear. Furthermore, our fear is flavored by the delicious expectation that the young lady will be saved in a heroic manner—an emotional response that derives from knowledge about the *form* of melodrama.

The phenomenon that Coleridge described can be seen to occur almost identically in drama and computer games, where we feel for and with the characters (including *ourselves* as characters) in very similar ways. Yes, someone might cry, but manuscripts and spreadsheets aren't pretend! Here we must separate the activity from its artifacts. The *representation* of a manuscript or spreadsheet as we manipulate it on the screen is in fact pretend, as compared to physical artifacts like data files (in the computer's memory or on a storage medium) and hard copy. The artifacts are real (as are actors, lighting

[6]For an analysis and thorough bibliography of Coleridge's criticism, see Allen and Clark [1962], pp. 221–239.

instruments, and reels of motion-picture film), but the rules involved in working with the *representations* (plays or human-computer activities) must subsume the knowledge, at some level, that they are representations. Why? First, because the fact that they are representations is the key to understanding what we can do with them. Second, because their special status as representations affects our emotions about them, enabling experiences that are, in the main, much more pleasurable than those we feel in real life. The distinguishing characteristic of the emotions we feel in a representational context is that there is *no threat of pain or harm in the real world.*

The key to applying the notion of "willing suspension of disbelief" to representational activities that have real-world artifacts is to ensure that the likelihood of *unintentional* effects on those artifacts approaches zero. The other day I experienced a power failure while I was working on this manuscript. I had learned to save my work often, but losing just a few paragraphs evoked plenty of unpleasant real-world emotion. Quite simply, *my system should never have let that happen.* My first word processor, although it lacked nearly all of the features that I appreciate in the one I use today, had a fail-safe feature that took the opportunity to automatically save an active file whenever there was a pause in the input stream—on the average, about every seven seconds. For people who use systems without such a feature, a power outage can be a context shift of the worst possible kind. Such interruptions to the flow of representational activity must be avoided if the powers of representational media are to be preserved. Saving my work has receded from an obsession to a kind of tic, but it shouldn't be there nipping at my subconscious at all.

Furthermore, engagement entails a kind of playfulness—the ability to fool around, to spin out "what if" scenarios. Such "playful" behavior is easy to see in the way that people use spreadsheets and word processors. In my house-buying example in the previous chapter, I played around with different scenarios for making trade-offs in my purchase decision. The key quality that a system must possess in order to

foster this kind of engagement is reversibility—that is, the ability to take something back. What if I failed to save a copy of my spreadsheet before I monkeyed around with a scenario that turned out to be disastrous? What if that scenario altered a significant amount of my data? The theory of hypertext suggests one solution, where various stages of a "document" (or, more correctly, an activity) can be saved and linked to the current version. This solution is unsatisfactory in that it is likely (at least in contemporary hypertext systems) to create a bewildering proliferation of documents. I don't really want to page back through versions of my work; I want to turn back the clock. The dimension of change is best represented through time, not fixed states. A simple chrono-scrollbar would suffice. Yes, the implementation is hard, but the hardest part is probably visualizing the appropriate representation in the first place.

I notice how word processing has changed my writing style. Now I am able to move chunks of text (roughly corresponding to ideas or elements in an argument) around within a document. I can more easily experiment with the visual components of the information I am creating by changing fonts and paragraph styles. But there is nothing sadder or more disruptive than seeing the message, `"Can't Undo."` With a typewriter, I still had the hard copy and a handy bottle of correcting fluid. Here again, the notion of document creation as an activity unfolding through time is superior to a notion of independent operations on an artifact of which one must remember to take snapshots.

Engagement is what happens when we are able to give ourselves over to a representational action, comfortably and unambiguously. It involves a kind of complicity. We agree to think and feel in terms of both the content and conventions of a mimetic context. In return, we gain a plethora of new possibilities for action and a kind of emotional guarantee. One reason why people are amenable to constraints is the desire to gain these benefits.

Engagement is only possible when we can rely on the system to maintain the representational context. A person should never be forced to interact with the system *qua* system; indeed, any awareness of the system as a distinct, "real" entity would explode the mimetic illusion, just as a clear view of the stage manager calling cues would disrupt the "willing suspension of disbelief" for the audience of a traditional play. Engagement means that a person can experience a mimetic world directly, without mediation or distraction. Harking back to the slogan, "the representation is all there is," we can see that interface designers are often engaged in the wrong activity—that is, representing what the *computer* is doing. The proper object of an "interface" representation is what the *person* is doing with the computer—the action. Thinking about things this way automatically avoids the trapdoors into meta-level transactions with "the system."

Characteristics of First-Person Experience

Another way to describe a person's involvement in the representational context of human-computer activity is as a *first-person* experience [see Laurel, 1986b]. In grammar, the personness of pronouns reflects where one stands *in relation to* others and the world. Most movies and novels, for example, are third-person experiences; the viewer or reader is "outside" the action and would describe what goes on using third-person pronouns: "First he did this, then they did that." Most instructional documents are second-person affairs: "Insert Tab A into Slot B"; "Honor your father and your mother." Operating a computer program is all too often a second-person experience: A person makes imperative statements (or pleas) to the system, and the *system* takes action, completely usurping the role of agency.

Agency is a key component of first-person experience. Although we may describe experiences in which we are not an agent using first-person pronouns (I saw this, I smelled that), the ability to *do* something sooner or later emerges as a criterion. On the one hand, doing very simple things can be an expression of agency—looking around, for instance, or reaching out and touching something (such simple types of

agency are often responsible for the "breakthrough" experiences reported by many people who have used contemporary virtual-reality systems). On the other hand, doing something relatively complex in an indirect or mediated way may not have a first-person feel. In the early days of computing, programmers would submit a program and data on punched cards and come back to pick up the results a day or two later. Although they were telling the computer what to do quite exactly, during the hours of waiting for the computer to "crunch" those programmers were not experiencing a feeling of agency. Today, imploring a system to do something in a highly constrained, formal language can engender a similar feeling that somebody (or something) else is in control.

This is not to say that people cannot experience agency when there are computer-based agents in the representational environment. Agents that are well characterized and amenable to dialogue and collaboration can give a person the sense of being one of several agents in a complex action. An agent can be a mentor or a dictator, a liberator or a jailor. The difference is in the person's experience of *agency*—the power to take action—whether the context includes other agents or not.

First-person sensory qualities are as important as the sense of agency in creating satisfying human-computer experiences. Quite simply, the experience of first-person participation tends to be related to the number, variety, and integration of sensory modalities involved in the representation. The underlying principle here is *mimetic*; that is, a human-computer experience is more nearly "first-person" when the activity it represents unfolds in the appropriate sensory modalities. The intuitive correctness of this notion is witnessed by the direction of technical evolution in the areas of simulators and games—toward higher resolution graphics and faster animation, greater sound capabilities, motion platforms, and mimetic input devices like force-feedback controllers. In task-oriented applications, new technologies are allowing researchers to replace indirect or symbolic representations and manipulations with direct, concrete ones—for example, physi-