5 The Product and Its Use Qualities

As we indicated in chapter 3, the development of a sense of quality and a language for articulating use-oriented qualities is a core element of interaction design ability. It is essential for designers to know what their products are and what they mean. The meaning of a product is never straightforward and unambiguous; it can never be obtained by the use of some objective scale of measurement. Of course, there are qualities that most people recognize and in some cases, there are even majority views on how they can and should be measured. In interaction design, such qualities typically include technical performance and structural features. However, most product qualities of interest to a designer are not that visible or easy to isolate. How do you measure usability and flexibility in a useful and practical way? How about the economical viability and ecological sustainability of a product? Even harder and less noticed are qualities such as social appropriateness, ethical justifiability, and aesthetic adequacy. Of course, researchers and designers have attempted to create means of measuring these and other qualities. In many camps, the ability to measure all relevant product qualities is seen as desirable. The intention is mainly to facilitate the design and deployment process; it can be safely stated, however, that there are no commonly agreed upon approaches for handling the more difficult aspects of digital artifact qualities.

The lack of objective measures does not mean that it is impossible for a designer to ponder product qualities—quite the contrary. A strong awareness and a set of powerful tools-for-thought are essential. A designer is never allowed to skip the question of product qualities by using the argument that nobody knows how these qualities can be measured. In any design situation, all qualities of the product will be determined whether they are measurable or not. It would be preferable if this determination was always the effect of a conscious and intentional design act, but in many cases the outcomes are unanticipated. We simply cannot consider all possibilities and safeguard ourselves against all unpredictable effects. There is only so much time and so many resources available to the design process, and the product ultimately has to be finished.

There are different kinds of design work, but the task for the designer always involves issues of needs, requirements, expectations, contexts, general trends and cultural traits, and even style and taste. A society and a market is always under the influence of a Zeitgeist—that is, a general and more or less shared set of views of what is structurally and culturally possible and acceptable in a specific design situation. It is important for the professional designer to be sensitive to these currents, but it is every bit as important to have a strong personal standpoint on what distinguishes a good product.

It may be worth pointing out in passing that knowing the qualities of everyday products does not necessarily mean knowing the qualities of digital artifacts. We are dealing with a strange material here, one where the spatial and the temporal meet in new ways. Much of our general sense of quality, what we know from handling the physical objects of everyday life, is not adequate when it comes to describing digital artifacts.

Our aim in this chapter is to provide tools-for-thought to help the reader build a sense of interaction design products and their qualities. In view of the inherent difficulties of objectively measuring the interesting qualities of digital artifacts, we advocate an approach based on *articulation*. This is to say that we view a designer's knowledge of product quality as an ongoing debate, a conversation with other designers and design theorists, as well as with design situations and the stakeholders involved in them. Statements are made in this debate through the main vehicles of design and reflection. A digital artifact or a design concept can be seen as a statement about a desirable product quality. Likewise for written or spoken analysis, where a core quality of a certain artifact genre or class of use situations is identified.

Product quality statements as outlined in this section are never generalized in the sense of straightforward application to new design situations. There is a significant amount of work involved in understanding the statement being made, assessing its relevance for the situation at hand, and figuring out what specific design actions it entails or supports (or discourages). By necessity, this work falls on the "reader" of the statement; the "writer" of the statement can make it easier for the reader by articulating the reasoning behind the statement and its possible scope and consequences.

Our approach in this chapter is therefore to provide the reader with a set of suggested, *use-oriented qualities of digital artifacts*. The qualities we propose are not general, but they have a scope of applicability that reaches across individual examples. One set of qualities concerns the users' *motivations* for engaging with the digital artifact, another addresses the immediate *sensation of interacting* with the artifact, and a third set has to do with the *social outcomes* of interaction. There is also a set of qualities pertaining to the *structural features* of the artifact as they manifest themselves in use and a final set addressing the induction of users' reflection upon their situation.

Our method is to illustrate most of the qualities we propose by analyzing concrete examples of digital artifacts in some depth. We want to emphasize the illustrative purpose of choosing these specific examples in the hope of making it clear to the reader not to focus too much on the chosen artifacts in terms of technical sophistication or state-of-the-art, nor as "killer applications" or classical artifacts. The examples are chosen to illustrate different *digital artifact genres* and the important qualities associated with them; they are not necessarily good examples in every respect. In fact, we doubt the existence of such examples since design is always a trade-off between different qualities and interests.

Here is a reasonable question the reader might have at this stage: "I want to be a game designer. Why should I read about the use qualities of an automatic teller machine (ATM)?" We can think of a few reasons why you should. First, a large part of the practical ability to design rests on having a repertoire of formats. As we indicated in chapter 3, formats are solution-oriented ideas, concepts, or examples. A broad collection of formats enables a designer to act more swiftly and confidently in new design situations. Secondly, the meta-skill of articulating product qualities and taking part in the ongoing debate about product quality is equally important in all genres of digital artifacts. Attempting to grasp examples of articulation work helps to build sensitivity to product quality, which is an essential part of design ability in any domain. Finally, and perhaps most importantly, the implicit notion of fixed genres and segmented markets is an obstacle to design innovation. We would argue that it is valuable for a game designer to know about the core qualities of ATM design, not in order to apply them uncritically to her own detailed design decisions, but rather to question and push the tacit boundaries of her work. Speaking of games and ATM design, here is an illustrative suggestion by Jeff Kipnis: "People like to play lotto and people like to use the ATM. Why don't you make it an option in the ATM to say put your money in and say, I'll bet a little bit and see if I can get a little more out, so you ask for twenty dollars, and you push the button, and you could get twenty-five or you could get fifteen" (qtd. in Dunne 1999, 54).

The specific idea may be good or bad, but it clearly has that unmistakable feeling of pushing a tacit boundary. The first time your read it, you might react with instant criticism: "You can't play games with people's hard-earned money! It wouldn't be . . . right." Then you read it again and start imagining what it would be like if your ATM played games with you. You find two days later that you still think about what an ATM-lotto machine would be like and how the idea could be used in other contexts. The tacit assumptions on ATM predictability and reliability have been pushed.

The qualities of digital artifacts we propose are summarized at the end of this chapter and related to other attempts to grasp the notion of quality in design. It should be noted here that the qualities we propose do *not* form a taxonomy. Even though our examples are chosen for their different core qualities, we maintain that the qualities we identify are interdependent in highly complex ways. The final quality and character of a particular digital artifact is *emergent* rather than additive.

A warning: Using the qualities we identify as a checklist for product evaluation would inevitably drain them of meaning. Instead, they should be seen as proposed tools for questioning, elaboration, and making informed choices in thoughtful interaction design. Design is always an act of composition, of shaping a whole and its parts simultaneously. The guiding principle in composition work is judgment of the emerging whole, as the complexity of design is too great for "divide-and-conquer" approaches. The main purpose of product quality articulation is to develop the ability to make such judgments, which constitute a thoughtful approach to understanding the qualities of digital artifacts.

5.1 Example: Automatic Teller Machines (ATMs)

An ATM consists of a computer, a small display screen, a numeric keypad, a card reader, and a cash dispenser. Some ATMs have another dispenser for statements and other printouts, and some feature unlabeled buttons around the display that take on different functionality in different dialogue states. The ATM computer has no internal database and relies on telecommunication with the computer at the current user's bank for balance information and other transactions. The user's ATM card contains the PIN code and identifies the bank. Keypad input is buffered in some situations—for instance, enabling the user to type the desired amount to withdraw before that question is presented on the display.

Using an ATM is like a conversation with somebody who likes to be in control of the dialogue:

ATM: Enter your PIN code!
Customer: 5555.
ATM: You can now withdraw cash, check your balance, or transfer money between your accounts.
Customer: Transfer money.
ATM: From which account?
Customer: 5555 313 2494.
ATM: To which account?
Customer: 5555 176 9921.
ATM: And how much do you want to transfer?
Customer: $100.

The most frequent services require only brief conversations. The buttons along the side of the screen are often used for withdrawal of preset amounts. The card is usually returned before the money and printouts. Some of the first ATMs were designed to dispense the money first, but in testing it was found that many people left their cards in the machine. The explanation is, of course, that the main goal was to withdraw money and when the money appeared, the goal was fulfilled and the card was forgotten.

It is easy to identify some shortcomings of the ATM design in terms of individual use properties, stemming from the rather old-fashioned infrastructure of transaction-based database communication. For instance the user has to go through several interaction steps to specify a transaction before the bank is contacted and the ATM may be notified that there is no money in the account. It would be straightforward (technically speaking) to redesign the ATM to initiate the user interaction and at the same time send for all information that might be needed at later stages of the interaction.

Another observation is that the interaction with an ATM is the same for everybody, even though we may have different uses for it. We already identify ourselves to the machine by inserting a more or less personal card. What if I could customize "my" ATM to offer a cash withdrawal of $50, with no receipt, at one press of a button? Other users might like to always have their current balance presented first, even though it would mean waiting for a few seconds after entering their PIN code.

5.1.1 Social Action Space

The most interesting aspects of the ATM are, however, concerned with issues that are larger than the design of the interface. It is clear that ATM technology illustrates a new way of handling money in everyday life. Before ATMs were invented, a person had to plan her cash needs in advance in order to make appropriate withdrawals while the bank was open. Now, all that person needs to do is make sure that she passes an ATM on her way to the restaurant. Some people rely on this kind of deferred planning to the extent that they depend on finding a working ATM. It is surely the case that a massive breakdown in telecommunications that disabled all ATMs in a city would show up quite clearly when stores in the city add up their sales for the day.

An ATM automates some of the simple bank customer services, makes the services more available to customers, and allows bank clerks to concentrate more on tasks that require skill and human judgment. When a customer walks inside the bank to withdraw an amount of cash, the clerk serves mainly as a mediator between the customer and the database. But to some customers, this mediation and human contact where the clerk personifies the bank is seen as very valuable. Would a bank lose customers if they referred all cash withdrawals to ATMs?

There is a significant trend in everyday life to reduce the use of cash. Some examples are credit cards, store cards, cash cards, postal money transfer, phone banking, and web banking. The main reason is that cash is expensive to handle. A customer transaction over the counter costs the bank roughly five times as much as via the web, and ATMs and manual phone services cost roughly two and a half times the web cost. Reducing the use of cash is also seen as a way to reduce robbery and other crime.

The point is that every product is designed with the (tacit or overt) intention of changing or facilitating change in the way people act. The ATM is a very clear example of how the social action space is designed. The intention is to change the activity patterns of bank customers from always coming into the bank to performing simple tasks by using machines. The design of fees for cash withdrawal over the counter compared to the ATM is also changed to further reinforce the activity-pattern change. However, ATMs also affect our social action space in other ways, some of which are unforeseen and perhaps undesirable. If a person is a deferred planner and finds the ATM closed at 9 P.M. on her way to meet up with friends at a bar, the situation may be characterized as a breakdown. It may in fact be the first time that this person notices the change in her own activity pattern.

It is not difficult for the designer to change users' social action spaces. Any change in the man-made environment, any new artifact, brings with it some kind of change in the social action space. The hard part is to predict the outcomes: the future social activities around the new artifact. For example, a new digital artifact may be designed with the intention of facilitating internal communication in an organization in order to overcome entrenchment and hostility. As it turns out, the new artifact is instead used as a forum for intense and upsetting debates, where employees anonymously voice unpleasant opinions about the organization and each other. This course of events may be good or bad, depending on the detailed context and on the point-of-view, but it is certainly clear that the artifact plays a significant role in shaping the activities in the organization. The social action space that is introduced was previously unthinkable, and apparently larger than the designers anticipated.

If we view changes in social action spaces as an important quality of (certain classes of) digital artifacts, then interaction design becomes more than the mere designation of a bundle of functions and gadgets. A more appropriate characterization might be that interaction design is the design of *conceivable social environments,* or as Terry Winograd puts it (qtd. in Preece, Rogers, and Sharp 2002, 70), "designing a space for people." The ATM is a good example of interaction design that has affected our social action space in many ways, with some expected outcomes and some surprising ones.

5.2 Example: Macromedia Director

Macromedia Director (referred to from now on as “Director”) originated as a tool for multimedia productions—that is, interactive presentations made up of text, image, sound, and video, and typically delivered on a CD-ROM. Even though it has expanded significantly, its original intentions are still discernible in its basic structures. Director is based loosely on the model of animated film editing, where a presentation (a “movie”) consists of a sequence of frames. All media elements to be presented in the movie frames are stored in one or more “casts.” The “score” is an editable representation of the frame sequence, and a basic movie is constructed by placing media elements from the cast into the score.

When the movie is played, each frame is presented with the media elements they contain. There is plenty of support for frame transitions—for instance, having a frame dissolve gradually into the next. Making media elements move is similar to cartoon animation; if you want to make a toy car drive across the stage from left to right, place it to the left in the first frame and move it a little to the right in each subsequent frame. A quicker way of getting the car to move is to indicate its positions at the start and the end of the sequence in “keyframes,” then have Director compute the intermediate positions in a process known as “tweening.”

Simple interactivity is also accommodated in the film-editing model. You can insert a “behavior” from a behavior library to make the playback pause on a certain frame. In that frame, a media element can be turned into a mouse-sensitive button with a behavior that moves the playback to another frame when clicked. Hypermedia information presentations, such as menu pages with related pages of content, are easy to construct in that manner.

In order to use Director for hypermedia work with simple structures and an emphasis on the multimodal content, the basic model described in the previous paragraphs is perfectly adequate. Historically speaking, that has also been Director’s main use: web site prototypes, adventure-style games with branching narrative structures, courseware, information presentations, and so on. But eventually, a designer is bound to come up against situations where the capacities of pure node-link hypermedia structures are too limited for the task at hand. It is at this point that Director’s underlying programming language Lingo becomes interesting.

The behaviors mentioned in this example are in reality small, encapsulated scripts; pieces of programming code, implemented in Director’s integrated scripting language Lingo. Lingo has grown more or less organically with each new version of Director into a fairly complete and general programming language, capable of much more than merely sending the playback head to specified frames in a score. Specifically, the two-dimensional and three-dimensional graphical primitives in Lingo are reasonably adequate and highly integrated with the film editing model and various drawing tools. Moreover, Lingo is interpreted, which means that any piece of code a programmer writes can be tested and debugged immediately, with no need for compile-link-load cycles. This makes Director (and Lingo) a useful choice for prototyping more advanced interaction techniques. Figure 5.1 shows some main elements of the Director and Lingo authoring environment.

The possibility of making a code template (a class) and creating multiple instances from it also gives Lingo some of the power of general object-oriented programming. A Lingo program created using such techniques does not need to be tied to a specific movie, but stays active in the computer's memory for as long as Director runs. We will not dwell on this subject; refer to Small (1996) for an introduction to the potential inherent in thinking of Director as a general programming environment rather than a multimedia production tool.

5.2.1 Transparency

Director and Lingo, when studied together, provide a good illustration of the quality of *transparency*. Starting to use Director for simple slide shows or website prototypes typically involves learning the film-editing model and working with the sequence of frames in the score. At this stage, Lingo is mostly used (unknowingly) in the form of encapsulated library behaviors to control the playback and perhaps some mouse-over effects. The designer only sees the system as the film-editing model.

As you move on to Lingo programming, the film-editing model becomes more transparent. The designer can see through to the language's underlying layers and finds that there is actually a fairly general programming environment behind the film-editing abstractions. Lingo code is not limited to controlling the playback of the score. One common insight that a budding Lingo programmer has is when she realizes that she can make an object move frame-by-frame not only by score-based animation, but just as easily by programming the motion into a piece of Lingo code. While learning to do this, the programmer typically finds that using Lingo to control the animation gives her much more flexibility and expressive power.

More generally, transparency can be seen as a continuum from the black box to the glass box (Rheinfrank, Hartman, and Wasserman 1992). A *black box* is a completely opaque artifact. The user provides input and the black box provides output, but there is nothing in the design of the black box to indicate what happens between input and output. The view of the black box becomes purely functional. In order to use it, all a person

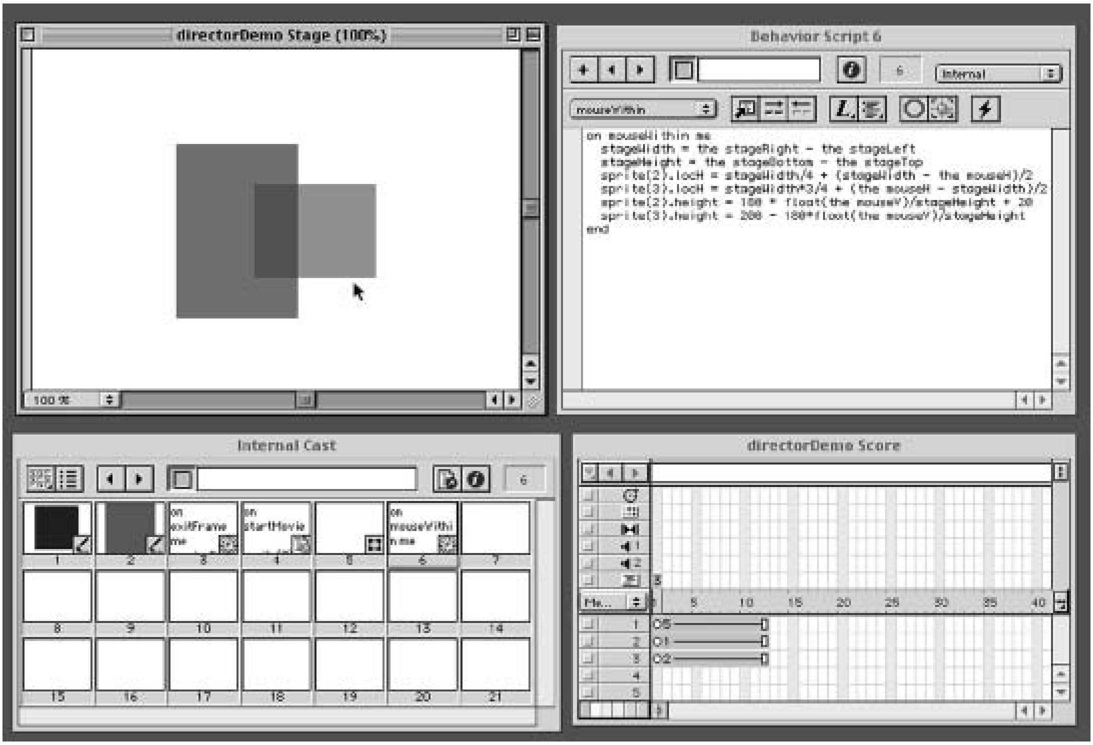

Figure 5.1
The most important parts of Macromedia Director (from the top left and clockwise): the stage, where the final program or presentation is set; the script window, where Lingo code is written; the score, which serves as a timeline for orchestrating playback; and the cast, which contains the elements of the presentation. This example shows a tiny interactive experiment, inspired by the work of Casey Reas, where the position and size of the two squares are related to mouse movements.

needs to know is the relation between what goes in and what comes out, between action and outcome; there is no way of knowing what goes on inside the black box.

The other extreme is the *glass box,* which is completely transparent. The user can see every detail of the artifact through the glass walls, and understand every part of its construction and every transformation in its information processing. The whole process is visible, which potentially leads to a better understanding of the artifact's implications, scope, and ways it can be changed.

Digital artifacts can occupy many possible positions along this continuum, from the totally opaque black box to the glass box where we can see all the way into the quantum physics of the transistors. The ATM is a black box, where the user only sees the relation between input and output, but not much in between. Another, slightly more complex example that exists between the black box and the glass box is the graphical user interface of a contemporary desktop operating system, such as Microsoft Windows. This system is transparent at the level of showing plainly that files are stored in hierarchical structures of directories. Anyone can drag a "document" from one "folder" to another, see it move, and understand the action to some extent. But the layers underlying this overt representation of the action are essentially opaque. For instance, it requires special tools and skills to learn how files are stored physically on the disk.

The designer of a digital artifact always has the difficult task of deciding what level of transparency to create in her design. In the ATM, for example, the intention is that the user should not be able to look inside the box. The ATM is a black box and this is probably a good choice for its purpose. Compared with the ATM, Director appears quite transparent, but in comparison with a traditional programming environment, it is still rather opaque. There are no means for controlling memory and register management, only rudimentary connections to the primitives of the underlying operating system, and so on.

The degree of transparency determines how the artifact can be used—that is, the nature of the resulting action space. An opaque design yields a smaller action space but greater degrees of security and control for the users, whereas a more transparent artifact is more flexible for the users but at higher risk: there is less control of how the artifact will be used and it is more vulnerable to interventions of different kinds.

Returning to the topic of design tools, less transparent designs may be preferable in many cases for novices in a certain field. Considering Director, the most important thing for the novice user to do is to become productive within the boundaries established by the program's outermost layer: simply structured multimedia work. The pleasure of reaching satisfactory results through a program like Director rapidly must not be underestimated. A common design strategy toward this goal is to provide examples, templates, and prepackaged effects for the user to start from, in addition to designing layers of transparency of the digital artifact in a conscious way.

A good example of the rapidly productive strategy is found in the area of bringing nonprofessional users in contact with the pleasures of creative media production. Or, to put it more plainly, helping people deal with their digital photography and home videos. There is a wide range of products offering well-designed templates and small sets of image-processing operations such as cropping, red-eye removal, and image effects. In a few easy steps snapshots of the family on vacation can be turned into nice-looking

Christmas cards. The transparency of these programs is very low—there are no settings for the filters, no general layout functions, no typographic control—and the productivity is instantly high. Image publication on personal web pages is another important niche for opaque and productive tools of this nature. Similar products are easy to find for home video editing as the computing power to capture and manage digital video is moving beyond the professional sphere of the movie business.

The paradigmatic example of rapidly productive tools in the field of three-dimensional computer graphics is Bryce, which was created by Kai Krause and colleagues at MetaCreations (of Kai's Power Tools fame) in the 1990s. The tools from MetaCreations were noticed mainly for their innovative interfaces; however, their main contribution in this context is the level of technical functionality they made accessible to a general audience on desktop computers. Highly automated and sophisticated 3-D modeling, manipulation, and rendering are hidden behind a skillfully designed layer of abstraction, enabling the nonprofessional user to concentrate on the visual results of 3-D graphics and produce stunning original results with little effort.

Unlike the Christmas card programs, Director addresses the issue of users evolving from simple experiments to demanding professional use. An advanced user requires more precision and expressive power from her tools. To meet these requirements, it is necessary to provide greater transparency and flexibility. The solution offered by Director is the underlying programming language Lingo, which is reasonably general and powerful. The general question the designer faces is whether to create a firm boundary between transparent and opaque, or whether to strive for a progressive disclosure of levels as the user's skills and needs develop. There are a few alternatives to consider if the boundary is not firm—for instance, whether the underlying levels of increasing flexibility and complexity should be hidden and require manual unveiling by the user. A trivial example would be the choice of abbreviated versus complete menus in a feature-rich word processor. Some experiments have been made with so-called adaptive interfaces where the underlying levels are hidden but the system discloses them based on independent reasoning about the user's skill level and needs.

The main point here is that all digital artifacts, like most technical systems, have the quality of transparency whether the designer wants them to or not. This is a necessary consequence of the complexity of the artifact. In many cases of interaction design, the quality of transparency is not intentionally designed. Instead, we suspect that it is common for interaction designers to act in accordance with what is considered to be the norm within a certain genre. Intentional and deliberate design requires an awareness of transparency as an important quality of digital artifacts.

5.3 Example: Feather

Feather is a concept designed by Rob Strong and Bill Gaver (1996) for personal communication in situations where one person travels while another stays at home. The aim of the concept is to indicate, simply and expressively, when the traveling partner is thinking of the other.

The system involves two devices: a picture frame carried by the traveler and a sculpture-like structure that remains with the person at home, with a transparent plastic cone containing a single feather and a hidden fan at the bottom. The traveler sends a signal to the partner at home by lifting the picture frame. The signal starts the fan and the feather rises to drift in the air inside the transparent cone, lifting and dipping as it catches the wind.

The Feather concept is exceptionally simple to describe, imagine, and build. It involves only one input and one output. Nevertheless, it has inspired a whole range of other designs and experiments in the emerging field of emotional communication. For instance, one aspect of Feather also identified by Strong and Gaver is the asymmetric nature of the communication. They have proposed a similar design called the "Shaker," where the two devices are identical in function: move one, and the other moves as well. This idea has been elaborated using other communication modalities, such as the White Stone concept: a squeeze of one soft, palm-sized stone heating the other stone slightly (Tollmar, Junestrand, and Torgny, 2000).

5.3.1 Personal Connectedness

In its simplicity, Feather illustrates a foundational quality of digital artifacts in their role as communication media: the possibility to stay in touch, to mediate closeness over a physical or temporal distance. When we think of connectedness in general, we tend to think of sending an email or an ICQ message, or talking to someone on a mobile phone. Feather opens our eyes to other modes of connectedness that are less intrusive and less demanding of our full attention, more subtle, and perhaps more poetic. The lifting of the picture frame gives the interaction a precious feel and the feather dancing in the air reflects the transience and lightness of thought.

There are interesting examples of how our rather crude and attention-demanding contemporary technologies for personal communication are put to innovative uses in search of a more subtle sense of connectedness. For a Scandinavian, the use of mobile phone SMS among young people instantly comes to mind. SMS stands for Short Message Service, a basic function of mobile phone networks where a text message of no more than 160 characters can be transmitted. Inputting text using a ten-key numeric keypad and navigating the message menus of mobile phones are tedious and error-prone tasks that would fail miserably in any proper usability test. Still, among Scandinavian teenagers the number of SMS messages sent significantly outnumbers mobile phone calls. In fact, the use of SMS has been one of the reasons why many Swedish teenagers find regular email unnecessary for their communication needs.

When an SMS message is received, a person's mobile phone typically beeps once. The message stays in her phone until she finds a convenient time and place to read it and possibly to reply to it. An interesting variation, apparently popular mainly among Italian teenagers, is the "drin" or "squillo," which means to dial somebody's mobile phone number, let it ring once, and then hang up. The recipient is aware of this communication code and does not pick up immediately. If the phone stops ringing after only one signal, someone has probably sent a drin. The point is that the call is stored in the phone's list of missed calls, to be viewed later as the equivalent of receiving a gentle thought through a Feather, Shaker or White Stone. Moreover, a non-answered call costs nothing to place or receive, while SMS is normally charged per message sent. This opens for a slightly different protocol in addition to the gentle-thought notion described earlier: If you receive a drin from a friend rather than from a loved one, it might also mean, "Call me, I have no more money on my cash card." Disambiguation is presumably based on personal knowledge of the receiver's relations with different drin senders.

The character of mobile phone communication, particularly among young users, becomes slightly more subtle and ambient, even though its crude interfaces and text-only protocols are still a long way from the interaction techniques illustrated by Feather. What these innovative appropriations of existing technology show is the need for communication not only at the center of our attention, but also on the periphery. There is more to communication than transmitting factual information.

The idea of using digital media as a means to support awareness of people's location and activities is by no means new. An early and influential example of this phenomenon is the work at Xerox Parc and EuroParc in the early 1990s on media spaces. The *in vivo* experiments were based on the idea of placing video and audio equipment in offices and common areas of office buildings that are possibly quite far apart from each other in geographical terms, but close in terms of collegiality, shared concerns, and joint projects. Applications were built for desktop computers that would provide users with peripheral awareness of colleagues in their offices, through thumbnail video presentations and continuous audio. Issues of privacy were identified and addressed by, for example, creating a multi-level protocol whereby a user could signal her availability to her colleagues. The combination of rather low-quality video and continuous audio were quite effective in creating a sense of a cohabited virtual space across geographical distance. Similar ideas have since been developed in several directions, including different forms of ambient displays as well as audio-only channels. The widespread use of instant messaging systems such as ICQ in recent years can be seen as another manifestation of the interest in experimenting with peripheral awareness, albeit adapted to more accessible low-bandwidth and low-technological conditions.

Digital infrastructure is spreading rapidly, with the explosion of the World Wide Web as one significant milestone and recent developments in wireless connectivity another possible milestone. In the digitized parts of the world, we are currently at a disturbing threshold level concerning social availability and awareness. It seems that our communicational practices are not quite adequate for the new infrastructure. Think about the frustration that could occur in a situation where a person sends an important message by email to someone she thinks will read it before the next day, then not getting an answer. At this point, she starts speculating on technical or personal reasons for the communication failure and finally ends up calling the person on the phone, only to learn that this person has taken a few days off. Or think about how something as private as a personal phone conversation or a confidential business discussion have, in a matter of a few years, turned into public performances, to the dubious pleasure of the fellow passengers in a train car, for instance.

We can think of the current communications situation as one where largely technology-led interventions have reshaped our social action space (cf. section 5.1) for personal communication in rather dramatic ways. Local practices evolving in different communities, such as the SMS and drin examples mentioned earlier, can be seen as collective experiments to explore the possibilities of the new social action space. Designers of digital artifacts for personal communication also contribute to the ongoing reshaping of the social action space. Compared to the sender of a drin, the position of the designer is more powerful: the designer's actions and decisions will affect the substrate, the infrastructure, and the potentiality of the action space.

The examples in this section indicate how almost any digital artifact can, and will, be used as a tool for communication and for creating connectedness. Following Dourish (2001), it might even be argued that digital artifacts have an intrinsic impact on personal connectedness (Dourish's first design principle of embodied computing states that computing is a medium). If this is the case, then personal connectedness becomes a more or less general quality in interaction design. Whenever a digital artifact is designed, the designer will affect the degree of connectedness among people as well as between people and artifacts, whether intentionally or unintentionally.

5.4 Example: An Interactive Visualization for Support and Maintenance Planning

A nuclear power reactor runs nearly all of the time. It is shut down for a few weeks out of every year for maintenance and repair work, and the number of tasks that need to be carried out in those weeks is staggering. Hundreds of external experts are hired to help out during this scheduled maintenance period. Security requirements are very high, for obvious reasons. It is important that every task can be traced and verified. Moreover, time is a critical resource. Every day that the reactor is down after the scheduled maintenance period is over costs the power plant large amounts of money in lost earnings.

In the middle of the 1990s, owners and operators of a nuclear power plant in Sweden were exploring the possibility of supporting the maintenance work by means of a workflow system. Previously, all maintenance was managed in a manual system, where work orders and other assignments were represented by paper forms to be signed when a task was completed. The idea was to model the work processes in the workflow system with descriptions of the necessary steps and responsible people. Each task should then be initiated from the computer, which would also provide the information needed to carry out the task.

The maintenance period at the nuclear power plant is planned long in advance and requires close monitoring and managing once it starts. A delayed task can have consequences for a whole series of other tasks. It may be necessary to reschedule tasks, reallocate resources, and make difficult priority judgments. Moreover, security and availability criteria must be fulfilled at all times. The shift supervisor is one of the central actors during the maintenance period. For the shift supervisor, an overall grasp of the situation is essential: what is going on right now, what is the status of critical tasks, and which tasks are planned to start shortly?

In 1996–1997, one of the authors (Löwgren) together with Martin Howard at Linköping University in Sweden designed a support system for maintenance management. The starting points for the design were the large amounts of information (tens of thousands of tasks during a scheduled maintenance period), the need to provide people responsible for certain tasks with a grasp of the whole situation, and our own desire to give the maintenance workers the means to control and plan ahead for their own work situation, rather than merely reacting to orders coming from the workflow engine.

Figure 5.2 illustrates the main ideas of a design concept. The situation is represented in a three-dimensional space with time on the z-axis running toward the observer. Every task is represented by a rectangular shape. The x-axis is divided into functional subsystems, so-called "system numbers," a term which is already a well-established part of

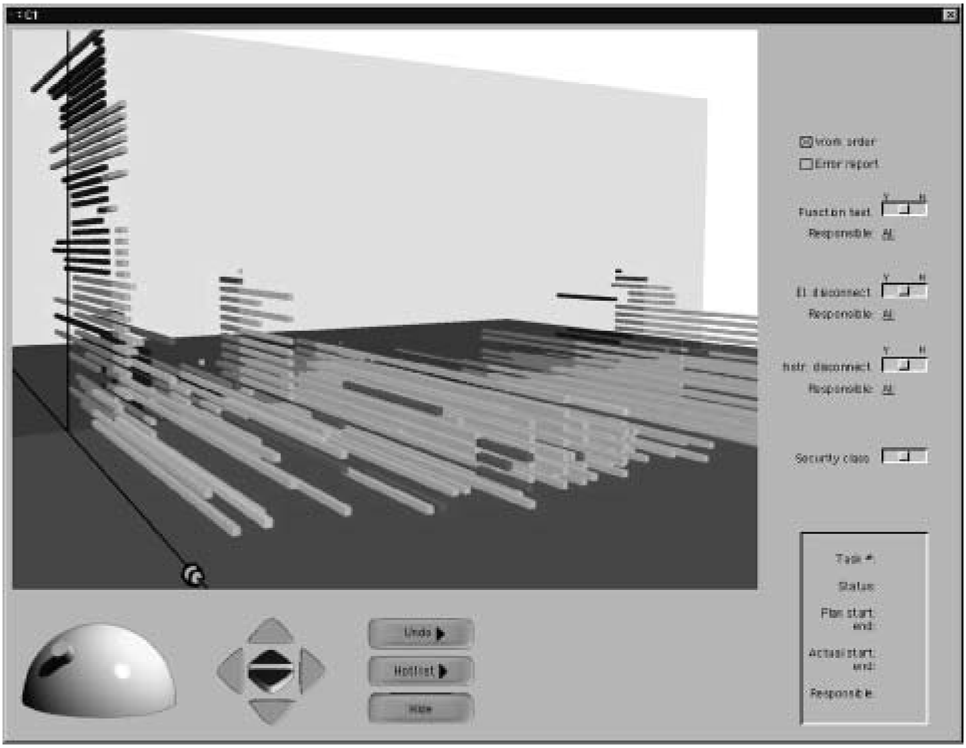

Figure 5.2
A design concept for managing large amounts of workflow information.

the work language at the site. The y-axis presents the number of tasks per subsystem. The semitransparent surface in the x-y plane represents the current time.

The controls to the right of the spatial presentation are filters that operate instantly on the presentation (see figure 5.3). The filter settings control which tasks are presented. For example, if a person only wants to see tasks planned to start tomorrow or later, she drags the planned start time filter to tomorrow's date and all tasks planned to start today or earlier instantly disappear. Filters are combined conjunctively, which means that the only tasks to be presented are the ones that pass through all the filter settings. Figure 5.3 illustrates an interaction sequence where the user first selects all tasks that contain function tests and then the tasks where the person with the initials SBH are responsible. The presentation in figure 5.3 shows that there are only three tasks that fulfil both selection criteria. The controls below the spatial presentation are used for navigation. Tasks can be viewed from different angles, at a distance for overview or close-up

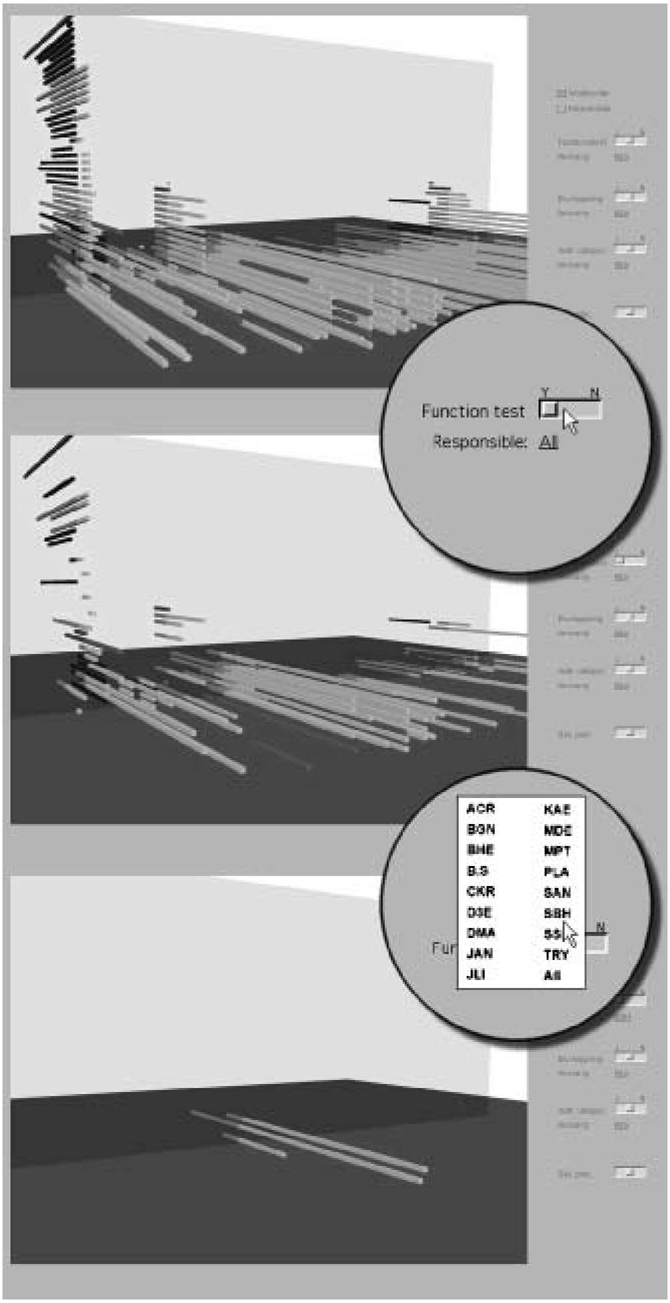

Figure 5.3
Filter settings instantly affect the presentation.

for details on a specific task. Movement is always animated to take the cognitive burden of view reconciliation off the viewer. Tasks can be temporarily hidden if they obstruct the line of vision. The "undo" function makes it possible to back out of an unsatisfactory series of navigation operations. Useful views and filter settings can be saved in a "hot list" and shared with colleagues.

5.4.1 Tight Coupling and Pliability

The information in the proposed maintenance management system corresponds to a relational database with a large number of records and several searchable keys. The filters are equivalent to search criteria for the different fields of records, combined with the logical conjunction "AND." Still, the concept is not very similar to a conventional design based on a database. (Figure 5.4 illustrates a more traditional version of a functionally similar system, with a search form, a summary of search results, and a detailed view of one of the records found.) The main conceptual difference is that this maintenance management database has been turned inside out. A regular database initially shows no contents. After formulating a search query, the user is presented with exactly the results that match the query. Our concept, on the other hand, initially shows all the data in the database when all the filters are disengaged. Activating different combinations of filters cuts away at the presentation until all that remains are the tasks the user needs to see for the moment.

The idea of turning databases inside-out was introduced by Ahlberg, Williamson, and Shneiderman (1992) who call the method "dynamic queries." A subsequent article (Ahlberg and Shneiderman 1994) introduces the use quality of *tight coupling*. The main idea is to minimize the distance between user intentions, user actions, and the effects of these actions. An example is the immediacy of filtering feedback. Any filter manipulation is instantly reflected in the presentation, and the users can gradually work their way toward the intended selection (or toward serendipitous discoveries, as is often the case with the visualization of large data sets). Other means of tightening coupling include providing a reliable undo function to back out of any undesirable interaction state and designing input controls that clearly show their current availability (for instance, graying out inapplicable buttons instead of presenting error messages after these buttons are clicked).

More generally, the quality sought in the interactive visualization described here can be identified as *pliability*. A set of information is pliable to the user if it feels like a responsive material that can be manipulated in an almost tactile sense. Pliability contributes to a highly involved process of exploration where the loop between senses, thought, and action is very rapid and physical rather than elaborate and mental. The

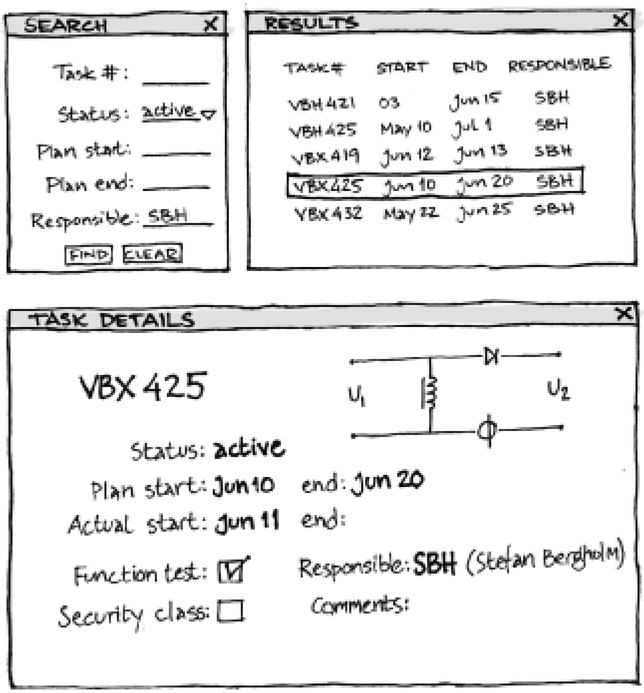

Figure 5.4
A functionally similar design for workflow information, shaped in a more traditional database form with search and presentation windows.

user makes a small quick move, the material shapes and responds, the user notices something new, she makes another move, and so on.

An example of exploring pliability on the micro-level of user-interface sensations is the Sens-A-Patch interaction technique for navigation of moderately sized information spaces (Löwgren 2001).

Sens-a-Patch (see figure 5.5) is based on the idea of spatial constancy—information elements stay in the place they are put on the navigation surface throughout the course of a session and across sessions. In order to fit many elements on to a small surface, the presentation is based on overlapping clusters, one of which is active at a time and the rest of which are visually faded into the background (but still legible). The user experience seems to create a certain amount of involvement, or at least visual or tactile interest. In one case where Sens-A-Patch was used to present all the contents of a medium-sized website, some users were observed to stay on the first page longer than their information needs dictated, in order to play with the sensation of navigating the

Figure 5.5
Navigating a Sens-A-Patch surface (starting from the top left). Clusters of information are activated and become more clearly visible as the cursor passes over them.

surface. This less goal-directed play at the first page may facilitate the serendipitous discovery of interesting contents.

Moving beyond the surface of the digital artifact, we find that pliability (as opposed to rigidity) is a possible direction in many fields of administrative data processing (Henderson and Harris 2000).[1] It is often the case that the use of administrative systems is unnecessarily restricted and constrained merely because of the underlying database structures used for implementation. A feasible alternative is to aim for more free-form data, basing disambiguation and other technical needs on social mechanisms as appropriate. A simple example is the rediscovery of the margins of paper forms, where annotations can be made and tied to the appropriate context (the form itself) for future interpretation. Most existing databases could easily be augmented with free-form fields similar in function to the margins of paper forms. Similar arguments can be made for the equivalent of sticky notes in "digital paperwork."

Pliability concerns the plasticity of digital "material." A pliable artifact supports exploration and fine-grained control; it is flexible to unanticipated needs and desires. In that sense, pliability can be seen as an attempt to articulate a design direction away from the rigidity that sometimes comes with traditional information systems, in terms of manual handling of the interface and work procedures embedded in the systems.

5.4.2 Control/Autonomy

Returning to the example of the maintenance support system at the nuclear power plant, it may seem odd to a reader skilled in workflow technology that the interactive visualization concentrates so strongly on providing users with the means to manage the maintenance information. Isn't the whole point of workflow systems to provide users with the task requests and other information they need as the workflow engine processes the predefined workflows?

The system design was based on a conscious choice to place a great deal of initiative with users and to offer a tool for their independent work planning and evaluation. The goal was to promote proactive use rather than reactive, which led to a general design principle of providing as much information as possible from the very beginning together with adequate means for accessing and managing it.

In general, what the user experiences as *control* is related to the degree of *autonomy* built into the digital artifact. A strongly autonomous design, an agent, is an artifact that acts on its own in a world defined by the symbols accessible to it. It maintains its own goals, chooses its own means, and can be said in some sense to have a will of its own. To the user, the agent is an actor who can be more or less collaborative.

In the 1990s, the idea of anthropomorphizing agents—endowing them with human traits or qualities—attracted some interest. The Knowledge Navigator future scenario from Apple in 1987 featured Phil, the intelligent desktop agent, who looks like a clever young man with a white shirt and bowtie. The user (a university professor) instructs him through spoken words to perform tasks such as answering the phone and finding all relevant unread articles for preparing a talk.[2] A more recent example of this desktop butler cliché is the character of the so-called Help function in Microsoft Office.

On the other end of the autonomy spectrum, purely nonautonomous artifacts are tool-like in their character. The user wields a tool to process materials and refine them into products of work. The tool is an extension of the hand or the eye—an instrument that facilitates or enables certain actions that remain under the strict control of the user.

The most interesting parts of the spectrum are, of course, between the two extremes of pure agent and pure tool. Virtual spaces are increasingly being used as habitats for artificial-life creatures where the user can affect the course of events to some extent. In some worlds, the user constructs her own creature and then returns later to learn how it has developed during its autonomous life in the virtual world. An example is “The Bush Soul” by Rebecca Allen (1997); artists such as Christa Sommerer and Jane Prophet have also presented work of this kind.

The genres of God-games and Sim-games can also be considered in terms of autonomy. An overall epic or a world simulation runs autonomously on a long-term time scale, where the player modifies local conditions and hopefully the general development of the game world by her actions. Are such virtual spaces and games autonomous or not? Clearly, they occupy places somewhere between the pure agent and the pure tool.

More mundane examples include the search function of a database system. It is an agent in the sense that even though it has been given instructions on a slightly abstract level as to what information to retrieve, it autonomously “chooses” how to carry out the task and which sectors on the disk to visit. However, it is also a tool, since it facilitates the processing of information material that the user would otherwise have to manage, search, and collate manually.

The question of appropriate positions on the autonomy spectrum has been subject to some debate, prompted by work on autonomous web agents in the mid-1990s. Norman (1994) argues for the primacy of user control over digital agents in order to eventually build user trust. If a user cannot trust an autonomous agent, it is of no use—even though it may have been designed to serve. When the user reads all the messages automatically marked for deletion by an email filter, just in case, then she does not save any time after all.

The maintenance support system at the nuclear power plant is an example of a use situation that designers approached with more or less tool-like ideals, where a more conventional workflow system would have been more of an autonomous agent. The general issue of agents versus interactive visualizations is addressed by Shneiderman and Maes (1997), who approach some sort of consensus in the proposition that agents may be suitable for various tasks that occur “behind the scenes,” but that the user must still be able to predict and control the externally visible properties of the system. They also note that the debate about positions on the autonomy spectrum sometimes compares apples and oranges, in the sense that visualizations are more appropriate for professional users and structured or semi-structured information spaces, whereas agents are useful mainly for intermittent use and unstructured information such as the World Wide Web. One might ask, however, how Norman’s requirement for user trust in agents relates to the idea of intermittent use.

As with the earlier qualities, control/autonomy is not always handled intentionally by interaction designers, even though all digital artifacts have this quality. Control/

autonomy has a significant impact on how people can and will use the digital artifact in question, and how the digital artifact will behave as an actor in a network of actors.

5.5 Example: Tetris

Tetris is perhaps one of the best-known computer games in existence. The basic idea behind it is extremely simple (see figure 5.6 for an illustration). Blocks of different shapes

Figure 5.6
The first Macintosh version of Tetris, released in 1988, running on a vintage Macintosh Plus 1 Mb with the nine-inch screen that this version of the game was designed for.

fall one at a time into a container, and the task is to align the blocks by rotation and sideways movements to fill the bottom of the container completely. Once a "layer" in the container is completely filled, it is automatically cleared to leave room for more blocks. Of course, the pace of the game increases as the player accumulates more points, and eventually the level of not-completely-filled layers reaches the top of the container. Game over.

The game of Tetris may be extremely simple, but it has also proven to be extremely addictive. It has been ported to nearly every computer, game console, and handheld computing device. In fact, most young users of information technology today probably think of Tetris as a typical puzzle game for the mobile phone. At one point, it was the best-selling computer game in the world and it is consistently rated highly on game critics' lists of outstanding classics. The core concept of the game has inspired thousands of clones and variations, including ones with alternative geometries, new block shapes, multiplayer possibilities, and additional functions and features. Given the game's long-standing popularity, it might be useful to sketch a brief outline of the history of Tetris.[3]

In 1985, the two computer engineers Alexey Pajitnov and Dmitry Pavlovsky at the Computer Center of the Moscow Academy of Sciences—who had experimented with making computer games on a mainframe computer—came in contact with high school student Vadim Gerasimov who knew how to program the IBM PC. They decided to team up and after a few months, Pajitnov came up with the original idea for Tetris based on an earlier game of his called "Genetic Engineering." This game in turn was inspired by the block shapes of a classical puzzle called Pentamino. The first prototype of Tetris was developed for the Electronica 60 computer, a Soviet clone of the PDP-11 with a monochrome character-based terminal where square brackets (like these: [ ]) were used to represent the falling blocks. It was then further developed on the IBM PC with colorful DOS graphics.

The creators discussed the possibilities of selling a collection of compelling PC games, including Tetris, but the system of business and governance in the former Soviet Union made this impossible. The game spread by word of mouth and personal contacts around Moscow, and somehow made it to Budapest where it was ported to the Apple II and Commodore 64 by Hungarian programmers. It was brought to the Western audience by Mirrorsoft and Spectrum Holobyte under unclear licensing conditions as the first game to emerge from behind the Iron Curtain, in a new version of the game for the IBM PC filled with Russian-themed graphics and music. It was an instant hit and became the best-selling game in the United Kingdom and the United States in 1988. The rest, as they say, is history.

5.5.1 Playability and Seductivity

If you have ever played Tetris, it is very likely that you have also said "Just one more time!" while staring at the game-over screen. This is a simple way of stating that Tetris exhibits a high degree of *playability* (Minter 1997). Terms like "addictive" are sometimes used to the same effect: to describe the enticing quality of a good game.

More generally, it is obvious that the use qualities of a game cannot be measured with the same yardstick as conventional productivity programs. We might claim that a new accounting system is good if it saves us half an hour that we can spend on more interesting work tasks, but to talk about a game that saves time would almost be a contradiction in terms. To the contrary, a game should be challenging and interesting enough for us to waste lots of time on it (from the point-of-view of our employer or family members).

The accounting program might be designed with the intention of making the user interface consistent and predictable in order to allow the user to concentrate on financial transactions rather than on handling the program. In games, on the other hand, the handling itself can be a vehicle for challenges that might occupy the player for days. This is perhaps best illustrated in an adventure game, where the current task in a traditional goal-oriented sense might be described as simply crossing the bridge. However, the interface is not designed to be transparent and facilitate the rapid execution of the task. To the contrary, there might be several interface puzzles that need to be solved before the bridge can be crossed and the "current task completed" (for example, find the hidden keyhole and figure out how to open it, insert the object that was lying around for no apparent reason in the bat cave three levels ago, activate the mechanism to open the bridge gate by figuring out the appropriate sequence of manipulations, and so on).

How, then, can we understand the mechanisms that make us say, "Just one more time" even at 2:00 A.M. when it is a workday tomorrow and we are already really tired? A crucial difference between games and most work-oriented productivity programs is that we engage with games because we want to, not because of external demands to perform or produce something. The rewards that motivate the game player do not come from the outside, but rather from the joy of playing or the sense of accomplishment involved in reaching a higher score, solving a mystery, or winning a tournament. In short, a game player is driven by intrinsic rather than extrinsic motivation.

Following Thomas and Macredie (1994), we can identify some of the elements contributing to intrinsic motivation.

Challenge The level of difficulty of the game increases with the player's proficiency.
Fantasy The player can experience magical events and perform magical actions.
Curiosity Information is tantalizingly hidden and requires effort to be revealed.

Novelty The player is continuously presented with new or transformed information and situations.
Complexity The game is difficult enough to require significant amounts of reasoning in order to make progress.
Surprise The behavior of the game and the unfolding of events are not easily predictable.
Control The player not only proceeds through and learns about the game through active participation, but also directs what is going on to some extent.
Competition The game offers possibilities to compete against oneself, the system, or other players—which provides the thrilling prospect of winning (or losing).

It is clear that the elements of intrinsic motivation are neither necessary in a strict sense, nor simply additive. For example, Tetris must be considered a success in terms of intrinsic motivation even though it draws almost exclusively on challenge, surprise, and competition with oneself. The elements of surprise and control are clearly in a trade-off relation to each other: something unpredictable is outside of player control. The point here is not to provide an epistemologically correct taxonomy, but merely to provide a basis for understanding playability as a use quality.

Another interesting approach is the concept of *seductivity,* which has been proposed as a way of understanding the captivating qualities of certain digital artifacts (Khaslavsky and Shedroff 1999). Seduction is described analytically as a process of enticement (attracting attention and making an emotional promise), relationship (making progress with small fulfillments and more promises, possibly lasting for a long time), and fulfillment (making good on the final promises and ending the experience in a memorable and positive way).

Khaslavsky and Shedroff offer the Visual Thesaurus by Plumb Design as an example of a seductive experience.[4] The Visual Thesaurus is a web application that adds new dimensions to the well-known contents of a traditional thesaurus by virtue of its interactive properties. Instead of database-style lookups, the user explores the synonyms of words and eventually the transient nature of language itself by navigating a beautifully animated network of words and their interrelations. In the analysis provided by Khaslavsky and Shedroff, the Visual Thesaurus offers surprising novelty for most users; goes beyond obvious needs and expectations; creates an emotional response due to its visual and interactional beauty; connects to personal goals through the fascination of words and concepts; promises to fulfill those goals; and leads the casual viewer to discover deeper meanings of looking up a word in the sense of the multidimensional and dynamic relationships between concepts.

It is straightforward to see how the notion of seductivity can be used to further our understanding of playability. For instance, a highly playable game might offer surprising novelty, create emotional responses through its visual and interactional qualities, allow for the formation of personal goals inside the game universe, and offer promises for the player to fulfill those goals.

The analysis in this section identifies qualities that distinguish good games, but does not say anything about how to achieve them. Designers and researchers have addressed the question of how to design a good game, and even though this chapter is not the place for an exhaustive discussion, we can provide a typical example. Pearce (1997) suggests that a game must contain a goal or objective known to all players from the outset; obstacles to create challenges for the player by impeding her progress towards the goal; resources to help the player achieve the goal and overcome obstacles; consequences in the form of rewards and penalties; information that is known to all players, to one player, or only to the game itself; and finally, rules and structure that provide the logical framework for the other elements of the game.

To open up another line of analysis, the classic status of the original Tetris game builds upon its character as a strongly challenging single-player competition with oneself. However, with the emergence of increasingly pervasive digital infrastructures comes the notion of a *game as a social activity,* of playing as doing something together with other people. Many successful single-player games have been developed into multiplayer versions. In fact, a two-player version of Tetris was designed already by Pajitnov and Gerasimov directly based on their work with the initial single-player version. In their two-player version of original Tetris, the container had no bottom. One player's blocks fell from the top down, the other's from the bottom up, and the object of the game was to compete for the space inside. Current plans at the Tetris Corporation, where Pajitnov is affiliated, include the development of software for Internet-based Tetris tournaments.[5]

Virtually all new games produced today—for consoles, personal computers, handheld computers, and mobile phones alike—involve at least some elements of network and social game play. Some are fundamentally social games, whereas most involve a combination of social elements and local, standalone features. A striking observation is that network games entail a new class of motivational factors, based on social interaction rather than on individual psychological considerations.

This is perhaps most obvious in role-playing games such as Everquest. Such games are typically based more or less directly on non-digital role-playing games, where a group of physically and temporally co-located players form a team and set out to perform quests in a fictitious world constructed and maintained by a game master. Or, to be more accurate, the role of the game master is typically to create the space of potentialities in the fictitious world. Its ongoing construction, maintenance, and elaboration is an act of collective narration where the players and the game master have equally important roles. The players act in accordance with their roles within the fiction; the game master takes care of acting for the characters the players meet, the overall unfolding of events unknown to the players, opponents' strategies and actions in battle, and so on. The main point of these games for our purposes here is that they build on social interaction within a fiction, but at the same time encourage a significant amount of social interaction outside the fiction that does not occur unless the fiction is present as a background. In other words, a group of role players can meet once a week for several years and have a great time involving plenty of interaction outside the fiction. But if the group decides to meet without playing the game, it is likely that they will find the conversations empty and something crucial missing.

In the digital versions of role-playing games, the role of the game master is typically automated and the players typically do not occupy the same physical space, but rather meet across physical distances and even from different countries. What emerges over time is a combination of two different game-playing cultures, or two different sets of motivations for playing. Many players engage in the game mainly for the pleasures of collaborative accomplishment (inside the fiction) and social interaction (inside and outside the fiction) together with old and new friends, all people they may never have met face to face. But there are also groups of players who engage in the game mainly to "win." Winning of course means different things in different games, but one good example is Everquest where the rules are based on the traditional family of (non-digital) fantasy role-playing games. A player's character gains points for accomplishments and the points translate into greater powers in the game, learning new skills, withstanding and being able to hand out more physical damage in battle, mastering more powerful magical spells, and so on. "Winning" translates roughly to attaining the highest possible level of player power. Strategies are plotted, alliances are forged, missions are planned—all with the intention of collecting as many points as possible. Everquest also contains a few "epic quests"—missions to accomplish inside the fiction—that somehow represent the ultimate level of achievement in the game universe.

It turns out that playability is a quality with many facets, including intrinsic motivation as well as social aspects. We may note that play, in a more general sense, is a concept with potentially broader scope than the teenage computer game genre. Many areas of everyday life contain aspects of game playing and role playing. Demarcations between work and play are becoming increasingly contrived, and it is in the interest of the thoughtful interaction designer to question them by means of reflective thought and design action.

5.6 Example: Signwave Auto-Illustrator

At first glance, Signwave Auto-Illustrator looks like any other vector-based drawing program. There is a tool palette, a drawing surface, property windows and so on. But once you start drawing, something unexpected happens. When you select the rectangle tool, drag out a rectangular shape on the drawing surface, and let go of the mouse button, the result is not the straight rectangle you would expect. What you get instead looks more like a child's drawing of a house. When you select the pencil and try to draw a curving line underneath the rectangle/house, the line turns into a cursive doodle when you release the mouse button. The text tool inserts nonsense words when you click to position the insertion point on the canvas. You select the house, open the color palette and try to choose a subdued pink color. The color palette would not let you do this, but taunts you for selecting lame colors and instead proposes a really strong, shocking pink. At this point, you start realizing that Auto-Illustrator is not like your average vector-based drawing program (see figure 5.7 for an illustration).

Auto-Illustrator is in fact a work of art, developed by Adrian Ward and awarded several prestigious prizes in the digital arts community. Clearly, it is also a piece of working software, a tool in the sense that it can be used to produce drawings.[6] But its character is much less submissive than conventional drawing programs. The basic shapes, such as rectangles and ellipses, are distorted by one of the autopilots in the program. The shape autopilot has different settings, and in the example above it happened to be set to childish style with average precision (which is between shabby and precise). The pencil autopilot choices range from insipid to cursive, and in the example from figure 5.7, it was set close to the cursive end of the scale. The text tool generates nonsense words with different linguistic traits (one setting is called slightly foreign, and uses numerous diacritical marks in the generated text). There is a tool to create a bug on the canvas—that is, a small dot that starts moving around on its own, leaving a visible trail behind it. There are tools and filters offering distortions one would not expect from a traditional tool, such as the instant Bauhaus style or the conversion of a vector drawing to a connect-the-dots exercise, complete with numbered dots. In short, Auto-Illustrator exercises a greater influence on the final drawings than a typical drawing tool would. The program exhibits a significant degree of autonomy;

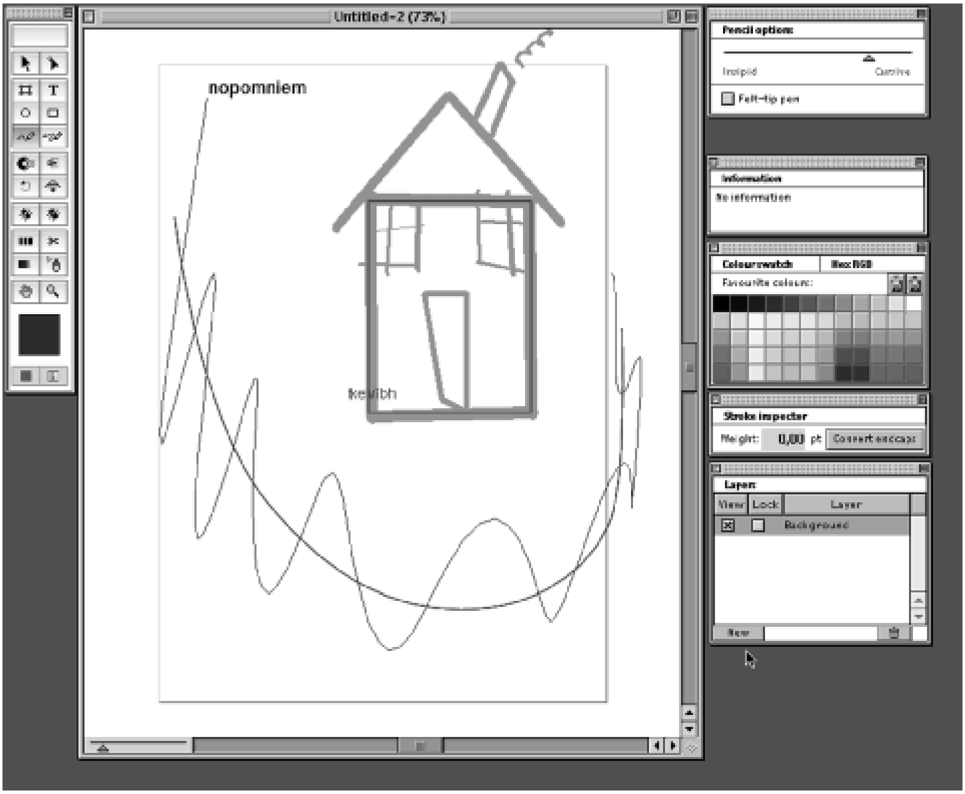

Figure 5.7
An Auto-Illustrator drawing in progress. The gray lines (rectangle and curve) are not part of the drawing; they are superimposed to illustrate the shapes that were actually drawn. (Image by permission of Signwave.)

the drawings are more clearly the results of a collaboration between the user and the program, or perhaps between the user and the designer of the program. In Adrian Ward's own words:

Auto-Illustrator is a semi-autonomous generative vector design application for Macintosh and Windows computers. While parodying existing professional graphic design software, Auto-Illustrator raises difficult questions and proposes new ways of interacting with artwork whose medium is software. As the user operates the software in order to produce a graphic design, the software interferes and makes its own decisions on how the design should look. The final design produced is no longer entirely the hand of the graphic designer, but also that of the software author, who has expressed himself through the use of code. Familiar questions of authorship and authenticity are raised again whilst in an entirely familiar environment—that of traditional desktop software.[7]

5.6.1 Parafunctionality

In the arts, including the digital arts, it is not uncommon for the artist to question what exists and to make the audience question their assumptions, prejudices, and everyday perceptions of life and reality. When this approach is applied to design, we might arrive at a design strategy that Dunne (1999) labels "critical design." In that context, he defines *parafunctionality* as a form of design where function is used to encourage reflection on our relationship with technology, or "how electronic products condition our behavior" (44). Using or attempting to use a parafunctional object creates a heightened sense of distance, mainly because it is conceptually difficult to assimilate into your view of reality. Acknowledging its usability or usefulness is hence also to discover new ways of seeing the world.

It should be noted that not all parafunctional objects can be used. A simple example is the Intolerable Object by Philippe Ramette whose lens would focus the sunlight directly onto the top of your head with possibly fatal consequences. Modeling a use scenario in your mind is in many cases enough to achieve the estrangement effect motivating the parafunctional design. The prerequisite for this to happen is that the proposed artifact is not too strange or else it will be immediately dismissed. It is a question of creating what Dunne and Gaver (1997) call a *value fiction*. If technology in science fiction is futuristic while the social values are conservative, the opposite is true in value fictions: The technologies are realistic while the social and cultural values are fictional or highly ambiguous. The viewer or imaginary user of a value fiction might ask herself why the values embodied in the proposal seem unreal and question the social and cultural mechanisms that define what is real in the first place (Dunne and Raby 2001).

It may seem that the quality of parafunctionality and, more generally, the practice of critical design, is mainly an artistic concern. In a narrow sense, this may be true, but we choose to include parafunctionality here since it illustrates an important aspect of the thoughtful stance. More than any other proposed use quality of digital artifacts, parafunctionality represents the distancing and critical reflection that is necessary for noticing the assumptions we normally take for granted about the role of digital artifacts in society and everyday life. This kind of thinking is every bit as necessary in the design of systems to support office work as it is in the arts.

5.7 An Incomplete Map of Digital Artifact Qualities

In figure 5.8, we have laid out a map of the eight use qualities discussed in the examples given so far. We have also introduced a handful of other qualities that we consider important for the broad picture. This map is by no means a comprehensive illustration of use qualities. Rather, we hope this map and our elaboration of the individual qualities

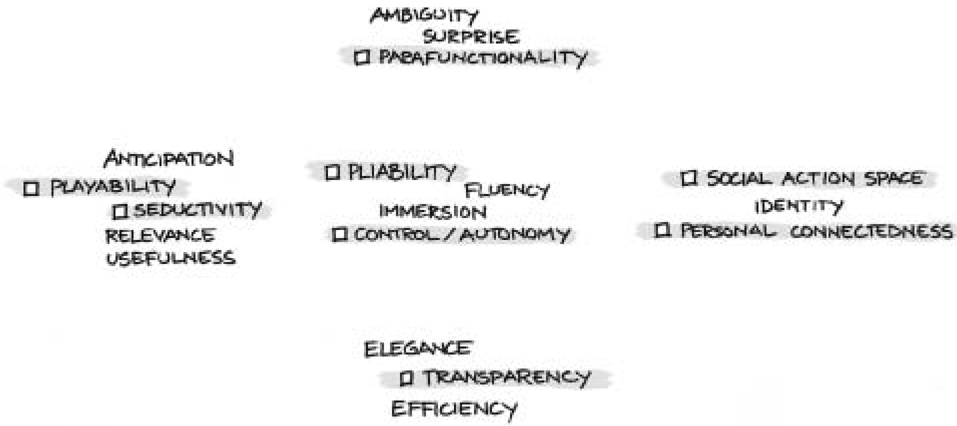

Figure 5.8
A map of the eight use qualities elicited from the examples in this chapter (highlighted) and ten more that we find important to discuss.

will inspire readers to make their own contributions and to pay attention to these and other use qualities in a more conscious way. The map is our way of grouping use qualities into a structure. Given such a structure, it might be easier to discover blank spots or recognize when too much emphasis is placed on certain aspects. The spatial layout of figure 5.8 does not carry any meaning in itself except that it structures and groups the qualities visually.

The group of qualities on the left of figure 5.8 deals with motivation Each of the qualities we propose is unique, but what they have in common is that they affect the user's motivation to continue using the artifact.

Playability is the addictive quality of a game that makes the player say "Just one more time!"

Seductivity is the emotional enticement of a digital artifact and its evolving relationship with the user.

Anticipation is a quality of use that has so far mainly been connected with dramatic structures and various forms of plot-driven interaction. In the context of interactive art, Fujihata (2001) describes the interaction process as one of participation and imagination:

> In an art of interactivity, one must be stimulated by interaction and enjoy having one's imagination activated. Interactivity is a stimulation of the power of imagination. By the power of imagination, one tries to see what will happen a few milliseconds ahead. This brings a future to the

present. It is a bridge between a past and a future. Only interactivity can make such a jump, enabling us to escape from the chronological cage. I believe it is a real creation.

Relevance and *usefulness* are examples of extrinsic motivational qualities, in the sense that the reasons for doing something may very well originate outside the user (even though the user may have internalized them to the extent that she thinks they are her own). Qualities such as these are inherited from work-oriented design of digital artifacts and from academic disciplines such as information systems and human-computer interaction. When a person calls something relevant, and even more so when she calls it useful, there is always the need to orient it toward a purpose: Useful for what? The traditional answer in the realm of digital artifacts is concerned with work tasks. If a system offers the information and tools a user needs to perform a task, then it is a relevant and useful system. The connections to modernist design notions such as "fitness for purpose" should be apparent.

Even though these concepts are typically used in reference to work tasks, it may be noted that the words in themselves do not preclude other applications. For instance, it seems quite sensible to talk about the relevance and usefulness of a website dedicated to fishing. But there are certainly some limits to relevance, usefulness, and other purpose-related qualities. Is Feather a relevant system? How useful is Tetris? At first, these questions may seem strange, but Feather might be relevant as a tool for keeping a friendship alive, and Tetris might be useful as a tool for relaxation or killing time. Such answers, however, border on stretching the concepts of usefulness and relevance beyond recognition.

The group of qualities in the middle of figure 5.8 deals with our immediate experience of interacting with a digital artifact, including our handling and perception of it

Pliability is the plasticity or malleability of the digital material in the hands of the user. Surface pliability is related to the tightness of the loop between the user making a move, perceiving the result, and understanding what she perceives. Deep pliability has to do with possibilities of acting freely and shaping the material, such as when the user annotates the margins of a form to communicate something outside the structured boundaries of the form itself.

Control/Autonomy deals with the distribution of initiative and responsibility in the interaction. Different degrees of a digital artifact's control/autonomy are conceivable, from pure tool to pure automaton.

Immersion deals with the handling of digital artifacts. At the focus of our attention, handling and perception of digital artifacts can become immersive. Digital artifacts offer possibilities for quasi-physical immersion through virtual reality technologies, where the idea is to fill our sensory organs as much as possible with the "virtual world." The virtual reality artwork Osmose, developed by Char Davies, is a powerful example, where the immersive effect comes from the program's exploiting of our kinesthetic sense of body and motion.[8] Moving around in the Osmose world is accomplished not by making contrived gestures with data gloves, but rather by the user breathing in and out and by shifting her body weight. Technically speaking, there is a sensor around the user's chest that is connected to her own vertical position in the virtual world. The user stands on sensors that are connected to speed and direction of travel. The immersive experience of navigating through the fundamental bodily function of breathing, however, is not reducible to simple technical understandings.

Immersion does not require expensive equipment or sensory-surround stimulation. There is another kind of immersion that comes from a person engaging so deeply in the task at hand that the world around it is forgotten. In terms of digital experiences, such immersion sometimes occurs in creative and explorative activities such as writing, drawing, playing games, or surfing the web. The experience is clearly related to the well-known psychological state of flow. A slightly more passive, but very real, form of immersion can come from being told a captivating story. Perhaps the most immersive activity in the digital realm, however, is programming where complex structures are built in the delicate balance between the programming language constructs and the limits of the programmer's own mental capacity.

Fluency as a use quality of digital artifacts is highlighted by the increasingly pervasive presence of digital infrastructure in our lives. "Use" is not necessarily on or off. It is rather more like a fluent dance among multiple representations. Information streams flow between center and periphery of our attention as we move through the shifting environments of everyday life and work. Transitions need to be graceful and undisruptive.

The group of qualities to the right of figure 5.8 concerns user's interactions with digital artifacts and their outcomes on a broader social level

Social action space is the potentiality for (social) action that is inherent in a digital artifact.

Personal connectedness is the quality of getting in touch, being in touch, and staying in touch with other people in a personally meaningful way. Note how personal connectedness is different from technical connectivity or availability, which deals with connections with little regard for who is connecting to whom and why.

Identity and the constructing and maintaining of identity is central in the use of digital artifacts, which possess symbolic use qualities like any other design objects. The recently emerging culture around skins for accessory desktop applications demonstrates our common desire to project just the right image. Translucent covers in organic shapes have been fitted onto every conceivable computing peripheral since the groundbreaking introduction of the iMac in 1998. But the construction of identity runs deeper than merely picking the right skin (whether it is made of pixels or plastic). The rapidly productive creative tools discussed in section 5.2 are important in this regard. A user with no training or innate talent in the visual arts can produce quite sophisticated results quickly and with little effort—and make a significant contribution to the ongoing project of reconstructing the user's image of herself.

The group of qualities at the bottom of figure 5.8 can be said to represent mediations of structural qualities, or engineering ideals as they are reflected in use qualities

Transparency in the sense we use it here has to do with the user's ability to uncover underlying layers of functionality and complexity as her learning proceeds or her needs evolve. The term is also used in human-computer interaction to indicate the unobtrusive ("intuitive") quality of a user interface that allows the user to concentrate on the task and the objects of the work domain. We do not make a strict separation between user interface and task domain, and rather see transparency as a dimension that can be consciously addressed in a design process.

Efficiency in using a digital artifact is typically connected to performing tasks for external purposes. Efficient use is rapid and error-free. One of the main forces behind the human-computer interaction field has historically been to improve the efficiency of computer-supported work tasks.

Elegance of a digital artifact, in a technical sense, is a combination of power and simplicity (Gelernter 1998). As a general aesthetic principle for engineering, an artifact should perform as well as possible with as simple of a construction as possible. For programming, this translates to creating elements (modules, objects, subroutines, or programs) that compute rapidly in few lines of source code. Note that simplicity itself is not necessarily a simple concept—a highly efficient and compact program can be almost impossible for anyone but a few experts to understand, but still be considered an elegant piece of work by virtue of its "power" and "simplicity."

Elegance in this technical sense is somehow related to the notion of functional minimalism, where the artifact is considered from the user's point-of-view and the goal is to offer appropriate core functions (power) and nothing else (simplicity). This can be seen as an engineering-aesthetic reaction to the exceeding amount of less-than-powerful features found in many mainstream applications. For instance, a word processor would be elegant if it were only good at word processing without also offering poor layout tools and even worse drawing tools.

The group of qualities on top of figure 5.8 deals with the user's creation of meaning in relation to a digital artifact

Ambiguity is generally considered detrimental in human-computer interaction, and it certainly stands in opposition to efficiency and transparency as those concepts are commonly interpreted. However, as Gaver, Beaver, and Benford (2003) argue, ambiguity can also be understood as a resource for encouraging close personal engagement with digital designs.

Gaver, Beaver, and Benford (2003) identify three types of ambiguity—information, context and relation ambiguity—and show how they have all been used to good effect in digital arts and design. One of their examples, Desert Rain, is a mixed-reality installation on the subject of virtual warfare and the blurring of boundaries between real and virtual worlds. The intention is to provoke participants to re-examine the boundaries between reality and fiction. To this end, the boundaries are deliberately ambiguous in that they mix elements of theater, performance, and computer game; the content is a mix of 3-D game-like graphics and video clips depicting real people's experiences of the Gulf War; rain curtains are used for projection, which provide a continually shifting and blurred view of the virtual world.

Ambiguity renders easy interpretation impossible by creating situations in which people are forced to participate in order to make some sort of meaning out of what they experience. An ambiguous design sets the scene for the creation of meaning, but does not prescribe an interpretation. The task of making the ambiguous situation comprehensible falls on the human actor, which may lead to inherent pleasure as well as a deeper conceptual appropriation of the design.

Parafunctionality is the quality of a digital artifact that encourages us to reflect on our relationship with technology, or more generally on the social and cultural values we hold and why we hold them.

Surprise is, of course, an element of parafunctional experience, but it also has interesting implications outside the realm of critical art and design. Holmlid (2002) discusses surprise as a use quality in relation to confusion in traditional work-oriented contexts, pointing out that the surprised user is interested in what she actually did (understanding the unexpected outcomes of an action), whereas the confused user is interested in what she should be doing instead of what she is doing. Surprise and confusion are not seen as errors but rather as natural parts of problem-solving activities, which might involve exploration of action possibilities inherent in the artifact as well as a reconsideration of the initial problem—the reason for encountering the surprise or confusion in the first place.

5.8 The Dynamic Gestalt

The eighteen use qualities we propose, and the structure of the map featured in figure 5.8, are based on our experiences and best understanding of digital design material. There is, however, one quality that we are aware of that we cannot capture in our map and it might be the most important quality of them all. The *overall character* of a digital artifact cannot be described by simply adding up a number of particular qualities. The artifact is more than the sum of its constituent parts; it has qualities that cannot be deduced from the structure and configuration of its parts—that is, it has holistic or emergent qualities. Figure 5.9 shows a very simple example, merely intended to illustrate the difference between additive and emergent properties.

Digital artifacts are every bit as temporal as they are spatial. In order to perceive the whole, or the *dynamic gestalt,* of a digital artifact, we need to experience it as a process, which is to say that we need to try it. The gestalt of a digital artifact *emerges in the interaction with the user over time.* There is no way for a user to get an idea of the dynamic gestalt without interacting with the artifact and exploring different possibilities and courses of events.

This means that the dynamic gestalt of a digital artifact can and must be described and analyzed as a whole, beyond the more particular use qualities we have introduced so far. For instance, interaction with a digital artifact has a temporal flow that can have different feels to it: calm, rapid, or stressful, for instance. Moreover, there is a dramatic structure to the dynamic process that spans across the course of the process from its introduction to its conclusion. This dramatic structure may, for instance, be described as inspiring, boring, obvious, or repetitive.

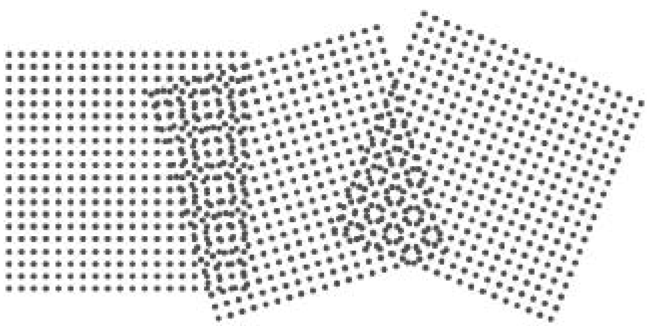

Figure 5.9
A simple visualization of emergent properties. The ring-shaped patterns where the rasterized squares interact are not easily predicted by looking at the squares in isolation, yet stand out clearly when the squares are superimposed. (Adapted from Davies and Talbot 1987.)

The examples of digital artifacts in this chapter are probably deeply unsatisfactory for a reader who has had no previous experience of using the artifacts described. The reader may get a sense from our way of writing that there is something interesting to be experienced in drawing with Auto-Illustrator, for instance, but our dry descriptions and screen shots will never be the same as using the program itself. This discrepancy is (hopefully) not a result of poor writing on our behalf, but rather an illustration of the dynamic gestalt concept. Auto-Illustrator, like any other digital artifact, has a dynamic gestalt that can only be apprehended by actually using it.

The dynamic gestalt of a digital artifact can be understood to some extent by the notion of overall *character.* We form ideas of people's (and artifacts') overall character rather quickly, ideas that are not deductively traceable to the sum of all individual actions and utterances of the person in question. Our idea of the character of a person is, in that sense, a holistic property. It is even the case that we use the notion of overall character to explain apparent inconsistencies in what we observe: "Oh, that is just part of his idiosyncratic character." The dynamic gestalt of a digital artifact is, in this sense, rather like its overall character (Janlert and Stolterman 1997).

There are other ways to characterize the dynamic gestalt of a digital artifact. A great deal of attention has, for instance, been paid to the idea of using metaphors as a means to describe digital artifacts, especially to describe their use qualities. On the other hand, we still find this to be an area that has not received enough attention. It seems as if the question of qualities has been mostly focused on single aspects of a digital artifact, particularly on aspects of an objectively measurable nature. The idea that the digital artifact has an overall character or gestalt that might overrule the effect of a single quality is a problematic—but particularly important—notion. Even though we still have no comprehensive way of characterizing the dynamic gestalt or overall character of a digital artifact, there is no excuse for not attempting to do so. Every interaction design will lead to a product, a digital artifact, that has a unique gestalt. Developing ways of describing, examining, criticizing and categorizing the overall character of such products should be a fundamental priority for our field and for anyone who wants to become a thoughtful designer.

5.9 Other Approaches to Design Quality

Attempts at creating languages to describe artifacts and their qualities are not unusual in traditional design disciplines, quite the contrary. Nearly every design discipline has its examples of such languages, whether they are presented as product semantics, design languages, or simply as the role of the critic in the larger knowledge-constructing system in which the designers and the artifacts also take part.

Krippendorff (1989) defines product semantics as "the study of the symbolic qualities of man-made forms in the cognitive and social contexts of their use and the application of the knowledge gained to objects of industrial design" (10). The primary concerns of product semantics are how artifacts make sense to their users, how they are symbolically embedded in society, and what roles they play in the ongoing self-production and reproduction of culture. Rheinfrank and Evenson (1996) take a designer's perspective on these same issues. They point out how what they call a "design language" can be consciously used by designers to communicate an understanding of intended artifact use to the users, create consistent and desirable images of, for example, a company through the design of its products, and affect a society's developmental trajectories.

A common notion seems to be that designers who are aware of product semantics and participate in the ongoing articulation of artifact qualities are capable of doing their work in qualitatively different ways. Articulation can be seen as a way to share and develop design knowledge, insights, and experiences among designers. Through articulation, designers and critics try to make explicit the qualities inherent in existing artifacts for assessment and appropriation. What we have presented in this chapter is a starting point for moving toward a more elaborate, intersubjective language for addressing the qualities of digital artifacts. Qualities such as the ones we have introduced must be assessed through dialogue, then elaborated, complemented, and possibly rephrased and combined into new formulations. This can be done by analyzing new kinds of digital artifacts and introducing the results into public debate.

This situation hints at the emergence of a new role in the knowledge-constructing system surrounding digital artifacts—namely, that of the *critic* (as introduced in chapter 4). The field of interaction design is well supplied with scientific evaluation of design concepts, and the creative development and dissemination of new design concepts within interaction design also works rather well. However, compared to other design disciplines, interaction design's lack of critics and criticism is obvious. There are occasional examples, such as the efforts by Johnson (1997), to situate the digital artifacts and communication media in a wider cultural context, but much more needs to be done to promote the vitality and progression of this field.

From a designer's point of view, however, there is still something missing. Even a very developed language for describing digital artifacts' use qualities does not in and of itself provide the necessary understanding of the totality created by these qualities as they are merged into a specific design. This totality comprises the basic structure of the artifact, all of its use qualities, and everything else pertinent to the artifact. In some cases, a certain use quality may be more or less irrelevant, whereas in other cases, it may dominate. The totality of a digital artifact is more than the sum of its constituent parts; it has desirable qualities such as flexibility, durability, and stability. A flexible and durable composition has a certain integrity and allows the user to make sense of it both immediately and as her relationship with it develops over time.

A language of use qualities says nothing about how to design an artifact or how to address its totality, but it may support the designer in her ongoing work of developing a repertoire, an articulation language, and a sense of quality. It may help the designer be prepared for new design situations, but it can never be a prescription for action in any specific situation.