Chapter 5

Understanding how interfaces affect users

5.1 Introduction

An overarching goal of interaction design is to develop interactive systems that elicit positive responses from users, such as feeling at ease, being comfortable, and enjoying the experience of using them. More recently, designers have become interested in how to design interactive products that elicit specific kinds of emotional responses in users, motivating them to learn, play, be creative, and be social. There is also a growing concern with how to design websites that people can trust, that make them feel comfortable about divulging personal information or making a purchase.

We refer to this newly emerging area of interaction design as affective aspects. In this chapter we look at how and why the design of computer systems cause certain kinds of emotional responses in users. We begin by looking in general at expressive interfaces, examining the role of an interface's appearance on users and how it affects usability. We then examine how computer systems elicit negative responses, e.g., user frustration. Following this, we present a debate on the controversial topic of anthropomorphism and its implications for designing applications to have human-like qualities. Finally, we examine the range of virtual characters designed to motivate people to learn, buy, listen, etc., and consider how useful and appropriate they are.

The main aims of this chapter are to:

- Explain what expressive interfaces are and the affects they can have on people.
- Outline the problems of user frustration and how to reduce them.
- Debate the pros and cons of applying anthropomorphism in interaction design.
- Assess the believability of different kinds of agents and virtual characters.
- Enable you to critique the persuasive impact of e-commerce agents on customers.

5.2 What are affective aspects?

In general, the term "affective" refers to producing an emotional response. For example, when people are happy they smile. Affective behavior can also cause an emotional response in others. So, for example, when someone smiles it can cause others to feel good and smile back. Emotional skills, especially the ability to express and recognize emotions, are central to human communication. Most of us are highly skilled at detecting when someone is angry, happy, sad, or bored by recognizing their facial expressions, way of speaking, and other body signals. We are also very good at knowing what emotions to express in given situations. For example, when someone has just heard they have failed an exam we know it is not a good time to smile and be happy. Instead we try to empathize.

It has been suggested that computers be designed to recognize and express emotions in the same way humans do (Picard, 1998). The term coined for this approach is "affective computing". A long-standing area of research in artificial intelligence and artificial life has been to create intelligent robots and other computer-based systems that behave like humans and other creatures. One well-known project is MIT's COG, in which a number of researchers are attempting to build an artificial two-year-old. One of the offsprings of COG is Kismet (Breazeal, 1999), which has been designed to engage in meaningful social interactions with humans (see Figure 5.1). Our concern in this chapter takes a different approach: how can interactive systems be designed (both deliberately and inadvertently) to make people respond in certain ways?

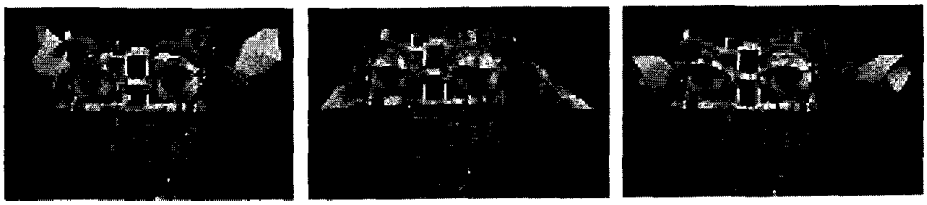


Figure 5.1 Kismet the robot expressing surprise, anger, and happiness.

5.3 Expressive interfaces

A well-known approach to designing affective interfaces is to use expressive icons and other graphical elements to convey emotional states. These are typically used to indicate the current state of a computer. For example, a hallmark of the Apple computer is the icon of a smiling Mac that appears on the screen when the machine is first started (see Figure 5.2(a)). The smiling icon conveys a sense of friendliness, inviting the user to feel at ease and even smile back. The appearance of the icon on the screen can also be very reassuring to users, indicating that their computer is working fine. This is especially useful when they have just rebooted the computer after it has crashed and where previous attempts to reboot have failed (usually indicated by a sad icon face—see Figure 5.2(b)). Other ways of conveying the status of a system are through the use of:

- dynamic icons, e.g., a recycle bin expanding when a file is placed into it
- animations, e.g., a bee flying across the screen indicating that the computer is doing something, like checking files
- spoken messages, using various kinds of voices, telling the user what needs to be done
- various sounds indicating actions and events (e.g. window closing, files being dragged, new email arriving)

One of the benefits of these kinds of expressive embellishments is that they provide reassuring feedback to the user that can be both informative and fun.

The style of an interface, in terms of the shapes, fonts, colors, and graphical elements that are used and the way they are combined, influences how pleasurable it is to interact with. The more effective the use of imagery at the interface, the more engaging and enjoyable it can be (Mullet and Sano, 1995). Conversely, if little thought is given to the appearance of an interface, it can turn out looking like a dog's dinner. Until recently, HCI has focused primarily on getting the usability right, with little attention being paid to how to design aesthetically pleasing interfaces. Interestingly, recent research suggests that the aesthetics of an interface can

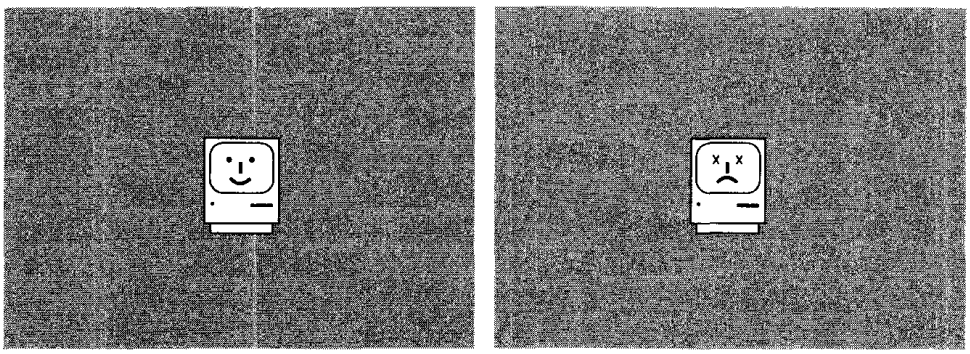


Figure 5.2 (a) **Smiling** and (b) sad **Apple** Macs.

have a positive effect on people's perception of the system's usability (Tractinsky, 1997). Moreover, when the "look and feel" of an interface is pleasing (e.g., beautiful graphics, nice feel to the way the elements have been put together, well-designed fonts, elegant use of images and color) users are likely to be more tolerant of its usability (e.g., they may be prepared to wait a few more seconds for a website to download). As we have argued before, interaction design should not just be about usability per se, but should also include aesthetic design, such as how pleasurable an interface is to look at (or listen to). The key is to get the right balance between usability and other design concerns, like aesthetics (See Figure 5.3 on Color Plate 6).

ACTIVITY 5.1 A question of style or stereotype? Figure 5.4 shows two differently designed dialog boxes. Describe how they differ in terms of style. Of the two, which one do you prefer? Why? Which one do you think (i) Europeans would like the most and (ii) Americans would like the most?

Comment

Aaron Marcus, a graphic designer, created the two designs in an attempt to provide appealing interfaces. Dialog box A was designed for white American females while dialog box B was designed for European adult male intellectuals. The rationale behind Marcus's ideas was that European adult male intellectuals like "suave prose, a restrained treatment of information density, and a classical approach to font selection (e.g., the use of serif type in axial symmetric layouts similar to those found in elegant bronze European building identification signs)." In contrast, white American females "prefer a more detailed presentation, curvilinear shapes and the absence of some of the more-brutal terms ... favored by male software engineers."

When the different interfaces were empirically tested by Teasley et al. (1994), their results did not concur with Marcus's assumptions. In particular, they found that the European dialog box was liked the best by all people and was considered most appropriate for all users. Moreover, the round dialog box designed for women was strongly disliked by everyone. The assumption that women like curvilinear features clearly was not true in this context. At the very least, displaying the font labels in a circular plane makes them more difficult to read than when presented in the conventionally accepted horizontal plane.

Another popular kind of expressive interface is the friendly interface agent. A general assumption is that novices will feel more at ease with this kind of "companion" and will be encouraged to try things out, after listening, watching, following, and interacting with them. For example, Microsoft pioneered a new class of agent-based software, called Bob, aimed at new computer users (many of whom were seen as computer-phobic). The agents were presented as friendly characters, including a friendly dog and a cute bunny. An underlying assumption was that having these kinds of agents on the screen would make the users feel more comfortable and at ease with using the software. An interface metaphor of a warm, cozy living room, replete with fire, furnishings, and furniture was provided (see Figure 5.5)—again intended to convey a comfortable feeling.

Since the creation of Bob, Microsoft has developed other kinds of agents, including the infamous "Clippy" (a paper clip that has human-like qualities), as part

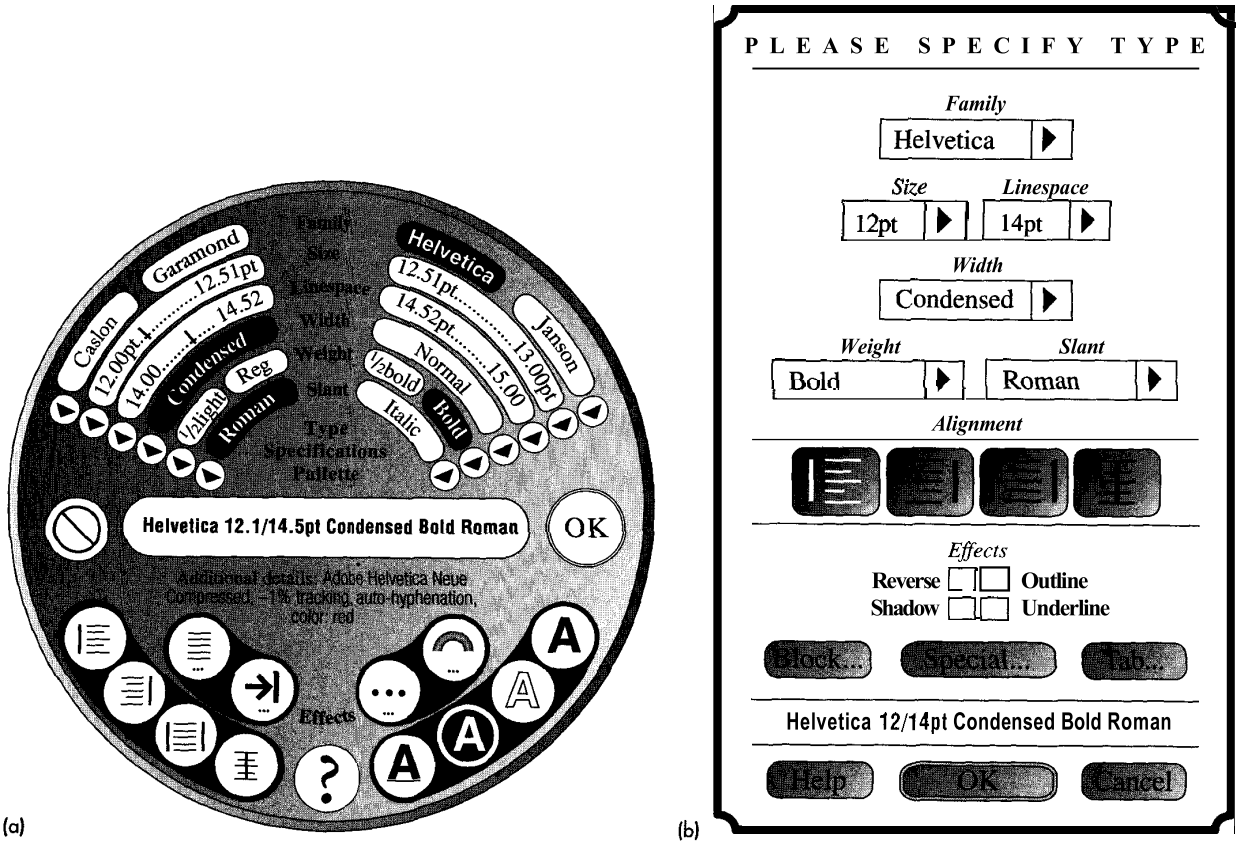


Figure 5.4 Square and round dialog boxes designed by Aaron Marcus (1993): (a) dialog box designed for white American women, and (b) dialog box designed for European adult male intellectuals.

Figure 5.5 "At home with Bob" software.

of their Windows '98 operating environment.' The agents typically appear at the bottom of the screen whenever the system "thinks" the user needs help carrying out a particular task. They, too, are depicted as cartoon characters, with friendly warm personalities. As mentioned in Chapter 2, one of the problems of using agents in this more general context is that some users do not like them. More experienced users who have developed a reasonably good mental model of the system often find such agent helpers very trying and quickly find them annoying intrusions, especially when they distract them from their work. (We return to anthropomorphism and the design of interface agents later in Section 5.5).

Users themselves have also been inventive in expressing their emotions at the computer interface. One well-known method is the use of emoticons. These are keyboard symbols that are combined in various ways to convey feelings and emotions by simulating facial expressions like smiling, winking, and frowning on the screen. The meaning of an emoticon depends on the content of the message and where it is placed in the message. For example, a smiley face placed at the end of a message can mean that the sender is happy about a piece of news she has just written about. Alternatively, if it is placed at the end of a comment in the body of the message, it usually indicates that this comment is not intended to be taken seriously. Most emoticons are designed to be interpreted with the viewer's head tilted over to the left (a result of the way the symbols are represented on the screen). Some of the best known ones are presented in Table 5.1. A recently created shorthand language, used primarily by teenagers when online chatting or texting is the use of abbreviated words. These are formed by keying in various numbers and let-

[1]On the Mac version of Microsoft's Office 2001, Clippy was replaced by an anthropomorphized Mac computer with big feet *and a hand that* conveys various gestures and moods.

Table 5.1 Some commonly used emoticons.

Emotion	Expression	Emoticon	Possible meanings
Happy	Smile	:) or :D	(i) Happiness, or (ii) previous comment not to be taken seriously
Sad	Mouth down	:(or :-<	Disappointed, unhappy
Cheeky	Wink	;) or ;-)	Previous comment meant as tongue-in-cheek
Mad	Brows raised	>:	Mad about something
Very angry	Angry face	>:-(	Very angry, cross
Embarrassed	Mouth open	:O	Embarrassed, shocked
Sick	Looking sick	:x	Feeling ill
Naïve	Schoolboyish look	<:-)	Smiley wearing a dunce's cap to convey that the sender is about to ask a stupid question.

ters in place of words, e.g., "I 1 2 CU 2nite". As well as being creative, the shorthand can convey emotional connotations.

Expressive forms like emoticons, sounds, icons, and interface agents have been primarily used to (i) convey emotional states and/or (ii) elicit certain kinds of emotional responses in users, such as feeling at ease, comfort, and happiness. However, in many situations computer interfaces *inadvertently* elicit negative emotional responses. By far the most common is user frustration, to which we now turn our attention.

5.4 User frustration

Everyone at some time or other gets frustrated when using a computer. The effect ranges from feeling mildly amused to extremely angry. There are myriads of reasons why such emotional responses occur:

- when an application doesn't work properly or crashes
- when a system doesn't do what the user wants it to do
- when a user's expectations are not met
- when a system does not provide sufficient information to let the user know what to do
- when error messages pop up that are vague, obtuse, or condemning
- when the appearance of an interface is too noisy, garish, gimmicky, or patronizing
- when a system requires users to carry out many steps to perform a task, only to discover a mistake was made somewhere along the line and they need to start all over again

ACTIVITY 5.2 Provide specific examples for each of the above categories from your own experience, when you have become frustrated with an interactive device (e.g., telephone, VCR, vending machine, PDA, computer). In doing this, write down any further types of frustration that come to mind. Then prioritize them in terms of how annoying they are. What are the worst types?

Comment In the text below we provide examples of common frustrations experienced when using computer systems. The worst include unhelpful error messages and excessive housekeeping tasks. You no doubt came up with many more.

Often user frustration is caused by bad design, no design, inadvertent design, or ill-thought-out design. It is rarely caused deliberately. However, its impact on users can be quite drastic and make them abandon the application or tool. Here, we present some examples of classic user-frustration provokers that could be avoided or reduced by putting more thought into the design of the conceptual model.

1. Gimmicks

Cause: When a users' expectations are not met and they are instead presented with a gimmicky display.
Level of frustration: Mild
This can happen when clicking on a link to a website only to discover that it is still "under construction." It can be still more annoying when the website displays a road-sign icon of "men at work" (see Figure *5.6).* Although the website owner may think such signs amusing, it serves to underscore the viewer's frustration at having made the effort to go to the website only to be told that it is incomplete (or not even started in some cases). Clicking on links that don't work is also frustrating.
How ***to avoid or help reduce the frustration:***
By far the best strategy is to avoid using gimmicks to cover up the real crime. In this example it is much better to put material live on the web only when it is complete and working properly. People very rarely return to sites when they see icons like the one in Figure 5.6.

2. Error Messages

Cause: When a system or application crashes and provides an "unexpected" error message.
Level of frustration: High
Error messages have a long history in computer interface design, and are notorious for their incomprehensibility. For example, Nielsen (1993) describes an early system that was developed that allowed only for one line of error messages. Whenever the

Figure 5.6 Men at work icon sign indicating "website under construction." According to AltaVista, there were over 12 million websites containing the phrase "under construction" in January 2001.

error message was too long, the system truncated it to fit on the line, which the users would spend ages trying to decipher. The full message was available only by pressing the PF1 (help key) function key. While this may have seemed like a natural design solution to the developers, it was not at all obvious to the users. A much better design solution would have been to use the one line of the screen to indicate how to find more information about the current error ("press the PF1 key for explanation").

The use of cryptic language and developer's jargon in error messages is a major contributing factor in user frustration. It is one thing to have to cope when something goes wrong but it is another to have to try to understand an obscure message that pops up by way of explanation. One of my favorites, which sometimes appears on the screen when I'm trying to do something perfectly reasonable like paste some text into a document, using a word processor, is: "The application Word Wonder has unexpectedly quit due to a Type 2 error."

It is very clear from what the system has just done (closed the application very rapidly) that it has just crashed, so such feedback is not very helpful. Letting the user know that the error is of a Type 2 kind is also not very useful. How is the average user meant to understand this? Is there a list of error types ready at hand to tell the user how to solve the problem for each error? Moreover, such a reference invites the user to worry about how many more error types there might be. The tone of the message is also annoying. The adjective "unexpectedly" seems condescending, implying almost that it is the fault of the user rather than the computer. Why include such a word at all? After all, how else could the application have quit? One could never imagine the opposite situation: an error message pops up saying, "The application has *expectedly* quit, due to poor coding in the operating system."

How to avoid or help reduce the frustration:

Ideally, error messages should be treated as how-to-fix-it messages. Instead of explicating what has happened, they should state the cause of the problem and what the user needs to do to fix it. Shneiderman (1998) has developed a detailed set of guidelines on how to develop helpful messages that are easy to read and understand. Box 5.1 summarizes the main recommendations.

BOX 5.1 Main Guidelines on How to Design Good Error Messages *(Adapted from Shneiderman, 1998)*

- Rather than condemn users, messages should be courteous, indicating what users need to do to set things right.
- Avoid using terms like FATAL, ERROR, INVALID, BAD, and ILLEGAL.
- Avoid long code numbers and uppercase letters.
- Audio warnings should be under the user's control, since they can cause much embarrassment.
- Messages should be precise rather than vague.
- Messages should provide a help icon or command to allow users to get context-sensitive help.
- Messages should be provided at multiple levels, so that short messages can be supplemented with longer explanations.

ACTIVITY 5.3 Below are some common error messages expressed in harsh computer jargon that can be quite threatening and offensive. Rewrite them in more usable, useful, and friendly language that would help users to understand the cause of the problem and how to fix it. For each message, imagine a specific context where such a problem might occur.

SYNTAX ERROR

INVALID FILENAME

INVALID DATA

APPLICATION ZETA HAS UNEXPECTEDLY QUIT DUE TO A TYPE 4 ERROR

DRIVE ERROR: ABORT, RETRY OR FAIL?

Comment How specific the given advice can be will depend on the kind of system it is. Here are suggestions for hypothetical systems.

SYNTAX ERROR—There is a problem with the way you have typed the command. Check for typos.

INVALID FILENAME—Choose another file name that uses only 20 characters or less and is lower case without any spaces.

INVALID DATA—There is a problem with the data you have entered. Try again, checking that no decimal points are used.

APPLICATION ZETA HAS UNEXPECTEDLY QUIT DUE TO A TYPE 4 ERROR—The application you were working on crashed because of an internal memory problem. Try rebooting and increasing the amount of allocated memory to the application.

DRIVE ERROR: ABORT, RETRY OR FAIL?—There is a problem with reading your disk. Try inserting it again.

3. Overburdening the user

Cause: Upgrading software so that users are required to carry out excessive housekeeping tasks

Level of frustration: Medium to high

Another pervasive frustrating user experience is upgrading a piece of software. It is now common for users to'have to go through this housekeeping task on a regular basis, especially if they run a number of applications. More often than not it tends to be a real chore, being very time-consuming and requiring the user to do a whole range of things, like resetting preferences, sorting out extensions, checking other configurations, and learning new ways of doing things. Often, problems can develop that are not detected till some time later, when a user tries an operation that worked fine before but mysteriously now fails. A common problem is that settings get lost or do not copy over properly during the upgrade. As the number of options for customizing an application or operating system increases for each new upgrade, so, too, does the headache of having to reset all the relevant preferences. Wading through myriads of dialog boxes and menus and figuring out which checkbox to

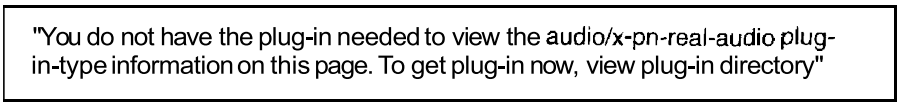


Figure 5.7a Typical message in dialog box that appears when trying to run an applet on a website that needs a plug-in the user does not have.

click on, can be a very arduous task. To add to the frustration, users may also discover that several of their well-learned procedures for carrying out tasks have been substantially changed in the upgrade.

A pet frustration of mine over the years has been trying to run various websites that require me to install a new plug-in. Achieving this is never straightforward. I have spent huge amounts of time trying to install what I assume to be the correct plug-in—only to discover that it is not yet available or incompatible with the operating system or machine I am using.

What typically happens is I'll visit a tempting new website, only to discover that my browser is not suitably equipped to view it. When my browser fails to run the applet, a helpful dialog box will pop up saying that a plug-in of X type is required. It also usually directs me to another website from where the plug-in can be downloaded (see Figure 5.7a). Websites that offer such plug-ins, however, are not organized around my specific needs but are designed more like hardware stores (a bad conceptual model), offering hundreds (maybe even thousands) of plug-ins covering all manner of applications and systems. Getting the right kind of plug-in from the vast array available requires knowing a number of things about your machine and the kind of network you are using. In going through the various options

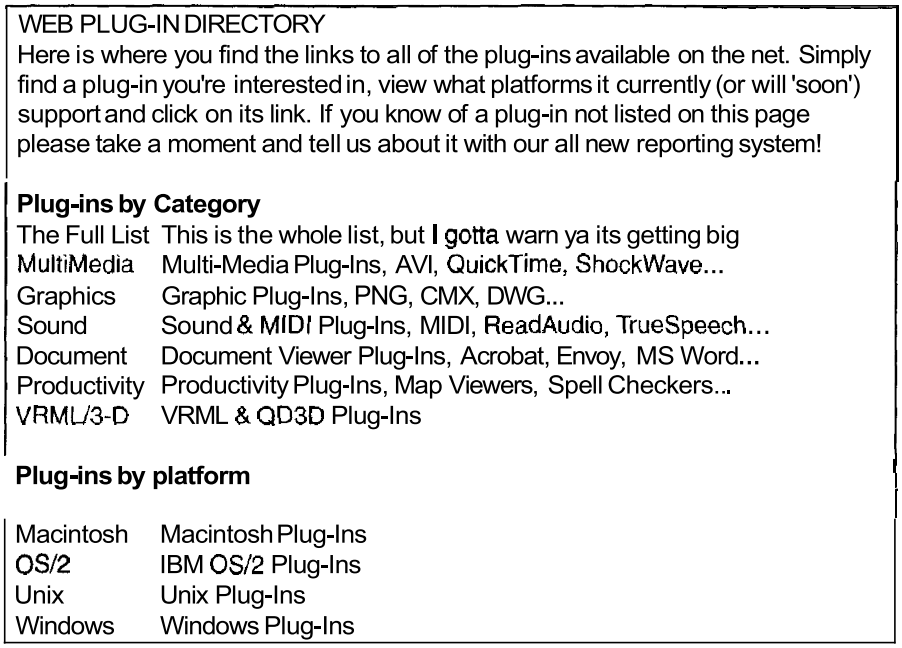


Figure 5.7b Directory of plug-ins available on a plug-in site directed to from Netscape.

to narrow down which plug-in is required, it is easy to overlook something and end up with an inappropriate plug-in. Even when the right plug-in has been downloaded and placed in the appropriate system folder, it may not work. A number of other things usually need to be done, like specifying mime-type and suffix. The whole process can end up taking huge amounts of time, rather than the couple of minutes most users would assume.

How to avoid or help reduce the frustration:

Users should not have to spend large amounts of time on housekeeping tasks. Upgrading should be an effortless and largely automatic process. Designers need to think carefully about the trade-offs incurred when introducing upgrades, especially the amount of relearning required. Plug-ins that users have to search for, download, and set up themselves should be phased out and replaced with more powerful browsers that automatically download the right plug-ins and place them in the appropriate desktop folder reliably, or, better still, interpret the different file types themselves.

4. Appearance

Cause: When the appearance of an interface is unpleasant
Level of frustration: Medium
As mentioned earlier, the appearance of an interface can affect its usability. Users get annoyed by:

- websites that are overloaded with text and graphics, making it difficult to find the information desired and slow to access
- flashing animations, especially banner ads, which are very distracting
- the copious use of sound effects and Muzak, especially when selecting options, carrying out actions, starting up CD-ROMS, running tutorials, or watching website demos
- featuritis—an excessive number of operations, represented at the interface as banks of icons or cascading menus
- childish designs that keep popping up on the screen, such as certain kinds of helper agents
- poorly laid out keyboards, pads, control panels, and other input devices that cause the user to press the wrong keys or buttons when trying to do something else

How to avoid or help reduce the frustration:

Interfaces should be designed to be simple, perceptually salient, and elegant and to adhere to usability design principles, well-thought-out graphic design principles, and ergonomic guidelines (e.g. Mullet and Sano, 1996).

5.3.1 Dealing with user frustration

One way of coping with computer-induced frustration is to vent and take it out on the computer or other users. As mentioned in Chapter 3, a typical response to seeing the cursor freeze on the screen is repeatedly to bash every key on the keyboard.

Another way of venting anger is through flaming. When upset or annoyed by a piece of news or something in an email message, people may overreact and respond by writing things in email that they wouldn't dream of saying face to face. They often use keyboard symbols to emphasize their anger or frustration, e.g., exclamation marks (!!!!), capital letters (WHY DID YOU DO THAT?) and repeated question marks (??????) that can be quite offensive to those on the receiving end. While such venting behavior can make the user feel temporarily less frustrated, it can be very unproductive and can annoy the recipients. Anyone who has received a flame knows just how unpleasant it is.

In the previous section, we provided some suggestions on how systems could be improved to help reduce commonly caused frustrations. Many of the ideas discussed throughout the book are also concerned with designing technologies and interfaces that are usable, useful, and enjoyable. There will always be situations, however, in which systems do not function in the way users expect them to, or in which the user misunderstands something and makes a mistake. In these circumstances, error messages (phrased as "how-to-fix-it"advice) should be provided that explain what the user needs to do.

Another way of providing information is through online help, such as tips, handy hints, and contextualized advice. Like error messages, these need to be designed to guide users on what to do next when they get stuck and it is not obvious from the interface what to do. The signaling used at the interface to indicate that such online help is available needs careful consideration. A cartoon-based agent with a catchy tune may seem friendly and helpful the first time round but can quickly become annoying. A help icon or command that is activated by the users themselves when they want help is often preferable.

BOX 5.2 Should Computers Say They're Sorry?

A provocative idea is that computers should apologize when they make a mistake. Reeves and Naas (1996), for example, argue that they should be polite and courteous in the same way as people are to one another. While apologizing is normal social etiquette in human behavior, especially when someone makes a mistake, would you agree that computers should be made to behave in the same way? Would users be as forgiving of computers as they are of one another? For example, what would most users think if, after a system had crashed, it came up with a spoken or written apology such as, "I'm really sorry I crashed. I'll try not to do it again"? Would they think that the computer was being sincere? Would the apology make them forgive the computer in the way they forgive other people, after receiving such an apology? Or would it have no effect at all? Worse still, would users perceive such messages as vacuous statements and regard them simply as condescending, thereby increasing their level of frustration? How else might systems communicate with users when they have committed an error?

5.5 A debate: the application of anthropomorphism to interaction design

In this section we present a debate. Read through the arguments for and against the motion and then the evidence provided. Afterwards decide for yourself whether you support the motion.

The motion

The use of anthropomorphism in interaction design is an effective technique and should be exploited further.

Background

A controversial debate in interaction design is whether to exploit the phenomenon of anthropomorphism (the propensity people have to attribute human qualities to objects). It is something that people do naturally in their everyday lives and is commonly exploited in the design of technologies (e.g., the creation of humanlike animals and plants in cartoon films, the design of toys that have human qualities). The approach is also becoming more widespread in interaction design, through the introduction of agents in a range of domains.

What is anthropomorphism? It is well known that people readily attribute human qualities to their pets and their cars, and, conversely, are willing to accept human attributes that have been assigned by others to cartoon characters, robots, toys, and other inanimate objects. Advertisers are well aware of this phenomenon and often create humanlike characters out of inanimate objects to promote their products. For example, breakfast cereals, butter, and fruit drinks have all been transmogrified *into* characters with human qualities (they move, talk, have personalities, and show emotions), enticing the viewer to buy them. Children are especially susceptible to this kind of "magic," as witnessed in their love of cartoons, where all manner of inanimate objects are brought to life with humanlike qualities.

Examples of its application to system design

The finding that people, especially children, have a propensity to accepting and enjoying objects that have been given humanlike qualities has led many designers into capitalizing on it, most prevalently in the design of human-computer dialogs modeled on how humans talk to each other. A range of animated screen characters, such as agents, friends, advisors and virtual pets, have also been developed.

Anthropomorphism has also been used in the development of cuddly toys that are embedded with computer systems. Commercial products like ActiMates™ have been designed to try to encourage children to learn through playing with the cuddly toys. For example, Barney attempts to motivate play in children by using human-based speech and movement (Strommen, 1998). The toys are programmed to react to the child and make comments while watching TV together or working together on a computer-based task (see Figure 1.2 in Color Plate 1). In particular, Barney is programmed to congratulate the child whenever he or she gets a right answer and also to react to the content on screen with appropriate emotions (e.g., cheering at good news and expressing concern at bad news).

Arguments for exploiting this behavior

An underlying argument in favor of the anthropomorphic approach is that furnishing interactive systems with personalities and other humanlike attributes makes them more enjoyable and fun to interact with. It is also assumed that they can moti-

vate people to carry out the tasks suggested (e.g., learning material, purchasing goods) more strongly than if they are presented in cold, abstract computer language. Being addressed in first person (e.g., "Hello Chris! Nice to see you again. Welcome back. Now what were we doing last time? Oh yes, exercise 5. Let's start again.") is much more endearing than being addressed in the impersonal third person ("User 24, commence exercise 5"), especially for children. It can make them feel more at ease and reduce their anxiety. Similarly, interacting with screen characters like tutors and wizards can be much pleasanter than interacting with a cold dialog box or blinking cursor on a blank screen. Typing a question in plain English, using a search engine like Ask Jeeves (which impersonates the well-known fictitious butler), is more natural and personable than thinking up a set of keywords, as required by other search engines. At the very least, anthropomorphic interfaces are a harmless bit of fun.

Arguments against exploiting this behavior

There have been many criticisms of the anthropomorphic approach. Shneiderman (1998), one of the best known critics, has written at length about the problems of attributing human qualities to computer systems. His central argument is that anthropomorphic interfaces, especially those that use first-person dialog and screen characters, are downright deceptive. An unpleasant side effect is that they can make people feel anxious, resulting in them feeling inferior or stupid. A screen tutor that wags its finger at the user and says, "Now, Chris, that's not right! Try again. You can do better." is likely to feel more humiliating than a system dialog box saying, "Incorrect. Try again."

Anthropomorphism can also lead people into a false sense of belief, enticing them to confide in agents called "software bots" that reside in chatrooms and other electronic spaces, pretending to be conversant human beings. By far the most common complaint against computers pretending to have human qualities, however, is that people find them very annoying and frustrating. Once users discover that the system cannot really converse like a human or does not possess real human qualities (like having a personality or being sincere), they become quickly disillusioned and subsequently distrust it. E-commerce sites that pretend to be caring by presenting an assortment of virtual assistants, receptionists, and other such helpers are seen for what they really are—artificial and flaky. Children and adults alike also are quickly bored and annoyed with applications that are fronted by artificial screen characters (e.g., tutor wizards) and simply ignore whatever they might suggest.

Evidence for the motion

A number of studies have investigated people's reactions and responses to computers that have been designed to be more humanlike. A body of work reported by Reeves and Nass (1996) has identified several benefits of the anthropomorphic approach. They found that computers that were designed to flatter and praise users when they did something right had a positive impact on how they felt about themselves. For example, an educational program was designed to say, "Your question makes an interesting and useful distinction. Great job!" after a user had contributed

a new question to it. Students enjoyed the experience and were more willing to continue working with the computer than were other students who were not praised by the computer for doing the same things. In another study, Walker et al. (1994) compared people's responses to a talking-face display and an equivalent text-only one and found that people spent more time with the talking-face display than the text-only one. When given a questionnaire to fill in, the face-display group made fewer mistakes and wrote down more comments. In a follow-up study, Sproull et al. (1996) again found that users reacted quite differently to the two interfaces, with users presenting themselves in a more positive light to the talking-face display and generally interacting with it more.

Evidence against the motion

Sproull et al.'s studies also revealed, however, that the talking-face display made some users feel somewhat disconcerted and displeased. The choice of a stern talking face may have been a large contributing factor. Perhaps a different kind of response would have been elicited if a friendlier smiling face had been used. Nevertheless, a number of other studies have shown that increasing the "humanness" of an interface is counterproductive. People can be misled into believing that a computer is like a human, with human levels of intelligence. For example, one study investigating user's responses to interacting with agents at the interface represented as human guides found that the users expected the agents to be more humanlike than they actually were. In particular, they expected the agents to have personality, emotion, and motivation—even though the guides were portrayed on the screen as simple black and white static icons (see Figure 5.8). Furthermore, the users became disappointed when they discovered the agents did not have any of these characteristics (Oren et al., 1990). In another study comparing an anthropomorphic interface that spoke in the first person and was highly personable (HI THERE, JOHN! IT'S NICE TO MEET YOU, I SEE YOU ARE READY NOW) with a mechanistic one that spoke in third person (PRESS THE ENTER KEY TO

Figure 5.8 Guides of historical characters.

BEGIN SESSION), the former was rated by college students as less honest and it made them feel less responsible for their actions (Quintanar et al., 1982).

Casting your vote: On the basis of this debate and any other articles on the topic (see Section 5.6 and the recommended readings at the end of this chapter) together with your experiences with anthropomorphic interfaces, make up your mind whether you are for or against the motion.

5.6 Virtual characters: agents

As mentioned in the debate above, a whole new genre of cartoon and life-like characters has begun appearing on our computer screens—as agents to help us search the web, as e-commerce assistants that give us information about products, as characters in video games, as learning companions or instructors in educational programs, and many more. The best known are videogame stars like Lara Croft and Super Mario. Other kinds include virtual pop stars (See Figure 5.9 on Color Plate 6), virtual talk-show hosts, virtual bartenders, virtual shop assistants, and virtual newscasters. Interactive pets (e.g., Aibo) and other artificial anthropomorphized characters (e.g., Pokemon, Creatures) that are intended to be cared for and played with by their owners have also proved highly popular.

5.6.1 Kinds of agents

Below we categorize the different kinds of agents in terms of the degree to which they anthropomorphize and the kind of human or animal qualities they emulate. These are (1) synthetic characters, (2) animated agents, (3) emotional agents, and (4) embodied conversational interface agents.

1. Synthetic characters

These are commonly designed as **3D** characters in video games or other forms of entertainment, and can appear as a first-person avatar or a third-person agent. Much effort goes into designing them to be lifelike, exhibiting realistic human movements, like walking and running, and having distinct personalities and traits. The design of the characters' appearance, their facial expressions, and how their lips move when talking are also considered important interface design concerns.

Bruce Blumberg and his group at MIT are developing autonomous animated creatures that live in virtual 3D environments. The creatures are autonomous in that they decide what to do, based on what they can sense of the 3D world, and how they feel, based on their internal states. One of the earliest creatures to be developed was Silas T. Dog (Blumberg, 1996). The **3D** dog looks like a cartoon creature (colored bright yellow) but is designed to behave like a real dog (see Figure 5.10). For example, he can walk, run, sit, wag his tail, bark, cock his leg, chase sticks, and rub his head on people when he is happy. He navigates through his world by using his "nose" and synthetic vision. He also has been programmed with various internal goals and needs that he tries to satisfy, including wanting to play

(a) (b) (c)

Figure 5.10 User interacting with Silas the dog in (a) physical world (b) virtual world, and (c) close-up of Silas.

and have company. He responds to events in the environment; for example, he becomes aggressive if a hamster enters his patch.

A person can interact with Silas by making various gestures that are detected by a computer-vision system. For example, the person can pretend to throw a stick, which is recognized as an action that Silas responds to. An image of the person is also projected onto a large screen so that he can be seen in relation to Silas (see Figure 5.10). Depending on his mood, Silas will run after the stick and return it (e.g., when he is happy and playful) or cower and refuse to fetch it (e.g., when he is hungry or sad).

2. Animated agents

These are similar to synthetic characters except they tend to be designed to play a collaborating role at the interface. Typically, they appear at the side of the screen as tutors, wizards and helpers intended to help users perform a task. This might be designing a presentation, writing an essay or learning about a topic. Most of the characters are designed to be cartoon-like rather than resemble human beings.

An example of an animated agent is Herman the Bug, who was developed by Intellimedia at North Carolina State University to teach children from kindergarten to high school about biology (Lester et al., 1997). Herman is a talkative, quirky insect that flies around the screen and dives into plant structures as it provides problem-solving advice to students (See Figure 5.11 on Color Plate 7). When providing its explanations it performs a range of activities including walking, flying, shrinking, expanding, swimming, bungee jumping, acrobatics, and teleporting. Its behavior includes 30 animated segments, 160 canned audio clips, and a number of songs. Herman offers advice on how to perform tasks and also tries to motivate students to do them.

3. Emotional agents

These are designed with a predefined personality and set of emotions that are manipulated by users. The aim is to allow people to change the moods or emotions of agents and see what effect it has on their behavior. Various mood changers are pro-

vided at the interface in the form of sliders and icons. The effect of requesting an animated agent to become very happy, sad, or grumpy is seen through changes to their behavior, For example, if a user moves a slider to a "scared" position on an emotional scale, the agent starts behaving scared, hiding behind objects and making frightened facial expressions.

The Woggles are one of the earliest forms of emotional agents (Bates, 1994). A group of agents was designed to appear on the screen that played games with one another, such as hide and seek. They were designed as different colored bouncy balls with cute facial expressions. Users could change their moods (e.g., from happy to sad) by moving various sliders, which in turn changed their movement (e.g., they bounced less), facial expression (e.g., they no longer smiled), and how willing they were to play with the other Woggles (See Figure 5.12 on Color Plate 7).

4. Embodied conversational interface agents

Much of the research on embodied conversational interface agents has been concerned with how to emulate human conversation. This has included modeling various conversational mechanisms such as:

- recognizing and responding to verbal and non-verbal input
- generating verbal and non-verbal output
- coping with breakdowns, turn-taking and other conversational mechanisms
- giving signals that indicate the state of the conversation as well as contributing new suggestions for the dialog (Cassell, 2000, p.72)

In many ways, this approach is the most anthropomorphic in its aims of all the agent research and development.

Rea is an embodied real-estate agent with a humanlike body that she uses in humanlike ways during a conversation (Cassell, 2000). In particular, she uses eye gaze, body posture, hand gestures, and facial expressions while talking (See Figure 5.13 on Color Plate 8). Although the dialog appears relatively simple, it involves a sophisticated underlying set of conversational mechanisms and gesture-recognition techniques. An example of an actual interaction with Rea is:

Mike approaches the screen and Rea turns to face him and says:

"Hello. How can I help you?"

Mike: "I'm looking to buy a place near MIT."

Rea nods, indicating she is following.

Rea: "I have a house to show you" (picture of a house appears on the screen). "It is in Somerville."

Mike: "Tell me about it."

Rea looks up and away while she plans what to say.

Rea: "It's big."

Rea makes an expansive gesture with her hands.

Mike brings his hands up as if to speak, so Rea does not continue, waiting for him to speak.

Mike: "Tell me more about it."

Rea: "Sure thing. It has a nice garden . . ."

ACTIVITY 5.4 Which of the various kinds of agents described above do you think are the most convincing? Is it those that try to be as humanlike as possible or those that are designed to be simple, cartoon-based animated characters?

Comment We argue that the agents that are the most successful are ironically those that are least like humans. The reasons for this include that they appear less phony and are not trying to pretend they are more intelligent or human than they really are. However, others would argue that the more humanlike they are, the more believable they are and hence the more convincing.

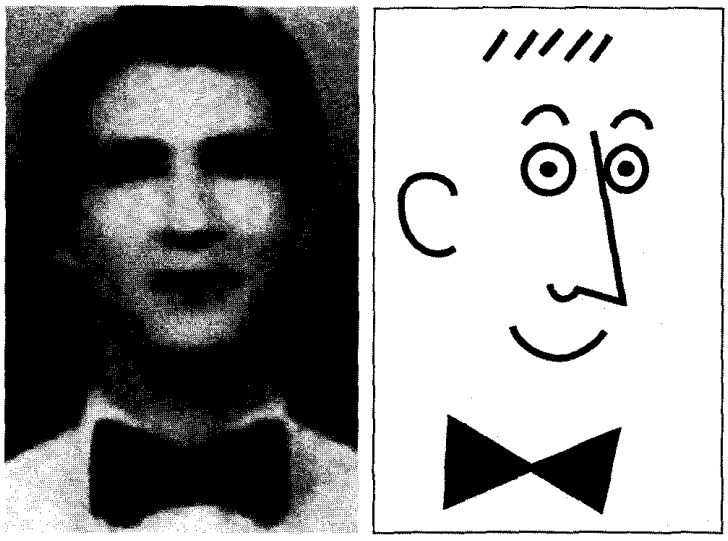
5.6.2 General design concerns

Believability of virtual characters

One of the major concerns when designing agents and virtual characters is how to make them believable. By believability is meant "the extent to which users interacting with an agent come to believe that it has its own beliefs, desires and personality" (Lester and Stone, 1997, p 17). In other words, a virtual character that a person can believe in is taken as one that allows users to suspend their disbelief. A key aspect is to match the personality and mood of the character to its actions. This requires deciding what are appropriate behaviors (e.g., jumping, smiling, sitting, raising arms) for different kinds of emotions and moods. How should the emotion "very happy" be expressed? Through a character jumping up and down with a big grin on its face? What about moderately happy—through a character jumping up and down with a small grin on its face? How easy is it for the user to distinguish between these two and other emotions that are expressed by the agents? How many emotions are optimal for an agent to express?

Appearance

The appearance of an agent is very important in making it believable. Parsimony and simplicity are key. Research findings suggest that people tend to prefer simple cartoon-based screen characters to detailed images that try to resemble the human form as much as possible (Scaife and Rogers, 2001). Other research has also found that simple cartoon-like figures are preferable to real people pretending to be artificial agents. A project carried out by researchers at Apple Computer Inc. in the 80s found that people reacted quite differently to different representations of the same interface agent. The agent in question, called Phil, was created as part of a promotional



(a) (b)

Figure 5.14 Two versions of Phil, the agent assistant that appeared in Apple's promotional video called the Knowledge Navigator (a) as a real actor pretending to be a computer agent and (b) as a cartoon being an agent. Phil was created by Doris Mitsch and the actor Phil was Scott Freeman.

video called "The Knowledge Navigator." He was designed to respond and behave just like a well-trained human assistant. In one version, he was played by a real actor that appeared on a university professor's computer screen. Thus, he was portrayed as an artificial agent but was played by a real human. The actor was a smartly dressed assistant wearing a white shirt and bow tie. He was also extremely polite. He performed a number of simple tasks at the computer interface, such as reminding the professor of his appointments for that day and alerting him to phone calls waiting. Many people found this version of Phil unrealistic. After viewing the promotional video, people complained about him, saying that he seemed too stupid. In another version, Phil was designed as a simple line-drawn cartoon with limited animation (see Figure 5.14) and was found to be much more likeable (see Laurel, 1993).

Behavior

Another important consideration in making virtual characters believable is how convincing their behavior is when performing actions. In particular, how good are they at pointing out relevant objects on the screen to the user, so that the user knows what they are referring to? One way of achieving this is for the virtual character to "lead" with its eyes. For example, Silas the dog turns to look at an object or a person before he actually walks over to it (e.g., to pick the object up or to invite the person to play). A character that does not lead with its eyes looks very mechanical and as such not very life-like (Maes, 1995).

As mentioned previously, an agent's actions need also to match their underlying emotional state. If the agent is meant to be angry, then its body posture, movements, and facial expression all need to be integrated to show this. How this can be achieved effectively can be learned from animators, who have a long tradition in this field. For example, one of their techniques is to greatly exaggerate expressions

and movements as a way of conveying and drawing attention to an emotional state of a character.

Mode of interaction

The way the character communicates with the user is also important. One approach has been towards emulating human conversations as much as possible to make the character's way of talking more convincing. However, as mentioned in the debate above, a drawback of this kind of masquerading is that people can get annoyed easily and feel cheated. Paradoxically, a more believable and acceptable dialog with a virtual character may prove to be one that is based on a simple *artificial* mode of interaction, in which prerecorded speech is played at certain choice points in the interaction and the user's responses are limited to selecting menu options. The reason why this mode of interaction may ultimately prove more effective is because the user is in a better position to understand what the agent is capable of doing. There is no pretence of a stupid agent pretending to be a smart human.

Assignment

This assignment requires you to write a critique of the persuasive impact of virtual sales agents on customers. Consider what it would take for a virtual sales agent to be believable, trustworthy, and convincing, so that customers would be reassured and happy to buy something based on its recommendations.

(a) Look at some e-commerce sites that use virtual sales agents (use a search engine to find sites or start with Miss Boo at boo.com, which was working at time of printing) and answer the following:

- What do the virtual agents do?
- What type of agent are they?
- Do they elicit an emotional response from you? If so, what is it?
- What kind of personality do they have?
- How is this expressed?
- What kinds of behavior do they exhibit?
- What are their facial expressions like?
- What is their appearance like? Is it realistic or cartoon-like?
- Where do they appear on the screen?
- How do they communicate with the user (text or speech)?
- Is the level of discourse patronizing or at the right level?
- Are the agents helpful in guiding the customer towards making a purchase?
- Are they too pushy?
- What gender are they? Do you think this makes a difference?
- Would you trust the agents to the extent that you would be happy to buy a product from them? If not, why not?
- What else would it take to make the agents persuasive?

(b) Next, look at an e-commerce website that does not include virtual sales agents but is based on a conceptual model of browsing (e.g., Amazon.com). How does it compare with the agent-based sites you have just looked at?

- Is it easy to find information about products?
- What kind of mechanism does the site use to make recommendations and guide the user in making a purchase?
- Is any kind of personalization used at the interface to make the user feel welcome or special?
- Would the site be improved by having an agent? Explain your reasons either way.

(c) Finally, discuss which site you would trust most and give your reasons for this.

Summary

This chapter has described the different ways interactive products can be designed (both deliberately and inadvertently) to make people respond in certain ways. The extent to which users will learn, buy a product online, chat with others, and so on depends on how comfortable they feel when using a product and how well they can trust it. If the interactive product is frustrating to use, annoying, or patronizing, users easily get angry and despondent, and often stop using it. If, on the other hand, the system is a pleasure, enjoyable to use, and makes the users feel comfortable and at ease, then they are likely to continue to use it, make a purchase, return to the website, continue to learn, etc. This chapter has described various interface mechanisms that can be used to elicit positive emotional responses in users and ways of avoiding negative ones.

Key points

- Affective aspects of interaction design are concerned with the way interactive systems make people respond in emotional ways.
- Well-designed interfaces can elicit good feelings in people.
- Aesthetically pleasing interfaces can be a pleasure to use.
- Expressive interfaces can provide reassuring feedback to users as well as be informative and fun.
- Badly designed interfaces often make people frustrated and angry.
- Anthropomorphism is the attribution of human qualities to objects.
- An increasingly popular form of anthropomorphism is to create agents and other virtual characters as part of an interface.

  People are more accepting of believable interface agents.
- People often prefer simple cartoon-like agents to those that attempt to be humanlike.

Further reading

TURKLE, S. (1995) ***Life on the Screen.*** New York: Simon and Schuster. This classic covers a range of social impact and affective aspects of how users interact with a variety of computer-based applications. Sherry Turkle discusses at length how computers, the Internet, software, and the design of interfaces affect our identities.

Two very good papers on interface agents can be found in Brenda Laurel's (ed.) ***The Art of Human-Computer Interface Design*** (1990) Reading, MA.: Addison Wesley:

LAUREL, B. (1990) Interface agents: metaphor with character, 355–366

OREN. T., SALOMON, *G.*, KREITMAN, K., AND DON. A. (1990) Guides: characterizing the interface, 367–381

MAES, P. (1995) Artificial life meets entertainment: lifelike autonomous agents. ***Communications of the ACM,*** 38. (11), 108–114. Pattie Maes has written extensively about the role and design of intelligent agents at the interface. This paper provides a good review of some of her work in this field.

Excerpts from a lively debate between Pattie Maes and Ben Shneiderman on "Direct manipulation vs. interface agents" can be found ***ACM Interactions Magazine,*** 4 (6) (1997), 42–61.