# Adaptive Interfaces and Agents

**Anthony Jameson**
DFKI, German Research Center for Artificial Intelligence
http://dfki.de/∼jameson

## INTRODUCTION

As its title suggests, this chapter covers a broad range of interactive systems. But they all have one idea in common: that it can be worthwhile for a system to learn something about each individual user and adapt its behavior to them in some nontrivial way.

An example that will be familiar to most readers is shown in Figure 22.1. A visitor to amazon.com has just explicitly requested recommendations, without having specified a particular type of product. During the user's past visits, AMAZON has learned about his interests, on the basis of items he has purchased and ratings he has made. Therefore, the system can make recommendations that are especially likely to appeal to this particular user.

*Concepts* The key idea embodied in AMAZON's recommendations and the other systems discussed in this chapter is that of *adaptation to the individual user*. Depending on their function and form, systems that adapt to their users have been given labels ranging from *adaptive interfaces* through *user modeling systems* to *software agents* or *intelligent agents*. Starting in the late 1990s, the broader term *personalization* became popular, especially in connection with commercially deployed systems. In order to be able to discuss the common issues that all of these systems raise, we will refer to them with a term that describes their common property explicitly: *user-adaptive systems*. Figure 22.2 introduces some concepts that can be applied to any user-adaptive system; Figure 22.3 shows the form that they take in AMAZON's recommendations.

A user-adaptive system makes use of some type of information about the current individual user, such as the products that the user has bought. In the process of *user model acquisition*, the system performs some type of learning and/or inference on the basis of the information about the user in order to arrive at some sort of *user model*, which in general concerns only limited aspects of the user (such as her interest in particular types of product). In the process of *user model application*, the system applies the user model to the relevant features of the current situation in order to determine how to adapt its behavior to the user.

*Note:* After some changes introduced by copy-editing, this chapter appeared in: A. Sears & J. A. Jacko (Eds.) (2008). *Human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (2nd ed.). Boca Raton, FL: CRC Press. It is almost entirely rewritten relative to the chapter from the first edition, which is available via the author's web homepage.


**Figure 22.1.** Part of a screen showing a list of recommendations generated on request by amazon.com. (Screen shot made from http://amazon.com/ in December 2005.)

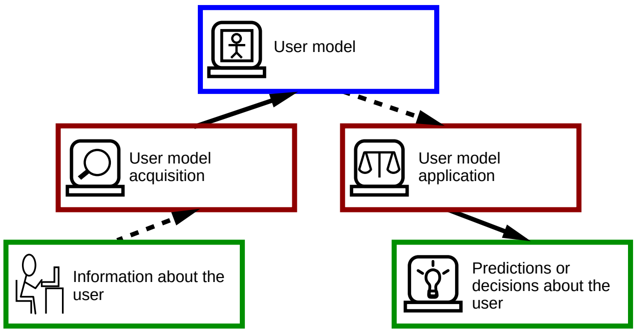


**Figure 22.2.** General schema for the processing in a user-adaptive system. (Dotted arrows: use of information; solid arrows: production of results.)

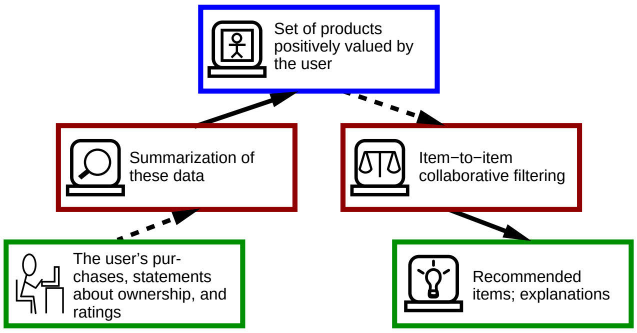


**Figure 22.3.** Overview of adaptation in amazon.com.

A user-adaptive system can be defined as:

- An interactive system that adapts its behavior to individual users on the basis of processes of user model acquisition and application that involve some form of learning, inference, or decision making.

This definition distinguishes user-adaptive systems from *adaptable* systems: ones which the individual user can explicitly tailor to her own preferences (for example, by choosing options that determine the appearance of the user interface). The relationship between adaptivity and adaptability will be discussed at several places in this chapter.

*Chapter Preview* The next two sections of this chapter address the question "What can user-adaptivity be good for?" They examine in turn ten different functions that can be served by user-adaptivity, giving examples ranging from familiar commercially deployed systems to research prototypes. The following section discusses some usability challenges that are especially important in connection with user-adaptive systems, challenges which have stimulated most of the controversy that has surrounded these systems. The next section considers a key design decision: What types of information about each user should be collected? The final major section looks at several distinctive aspects of empirical studies of user-adaptive systems. The chapter concludes with comments on the reasons why their importance is likely to continue to grow.[1]

## FUNCTIONS: SUPPORTING SYSTEM USE

Some of the ways in which user-adaptivity can be helpful involve support for a user's efforts to operate a system successfully and effectively. This section considers five types of support.

[1]The version of this chapter in the first edition of this handbook (Jameson, 2003) included a section about some of the machine learning and artificial intelligence techniques that are most commonly used for user model acquisition and application. There is no such section in this second edition, because (a) it seemed more important to expand the material in the other sections and (b) the range of techniques used has grown to the point where a brief summary would have limited value. Discussions of the relevant methods will be found in many of the works cited in the chapter.

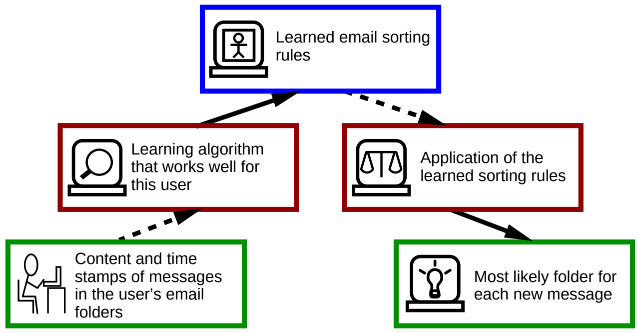


**Figure 22.5.** Overview of adaptation in I-EMS.

### Taking Over Parts of Routine Tasks

The first function of adaptation involves taking over some of the work that the user would normally have to perform herself—routine tasks that may place heavy demands on a user's time, though typically not on her intelligence or knowledge. Maybe the most obvious task of this sort is organizing email, which takes up a significant proportion of the time of many office workers. This was one of the tasks addressed by the classic early work of Pattie Maes's group on "agents that reduce work and information overload" (see, e.g., Maes, 1994).

A more recent effort (Figure 22.4) is found in the prototype "intelligent electronic mail sorter" I-EMS (McCreath, Kay, & Crawford, 2005; see also Crawford, Kay, & McCreath, 2002; McCreath & Kay, 2003) which is designed to expedite the tedious task of filing incoming email messages into folders. By observing and analyzing the way an individual user files messages, the system learns to predict the most likely folder for any new message. In the overview of messages in the user's inbox (shown in the top part of the screen shot), I-EMS tentatively sorts the new messages into categories that correspond to the most likely folders. When an individual message is being displayed, the one-line field in the middle of the screen shows an explanation of the folder prediction. If the user agrees with the prediction, she can click on the "Archive" button at the top of the screen to have the message moved into the predicted folder; to file it away in another folder, she drags it to the icon for the folder in the left-hand panel, just as she would with a system that did not make any predictions.

One reason why research on systems like I-EMS has continued for so long is that the problem raises a number of challenges. For example, since different users apply radically different principles for creating categories of email, I-EMS supplies several different methods for learning a user's implicit rules, each of which may show a different degree of success with different users. It also allows the learned rules to operate alongside any hand-crafted rules that the user may have defined, so that the strengths of both types of rule can be exploited (an evaluation is discussed below in the section on empirical methods). And since even the best

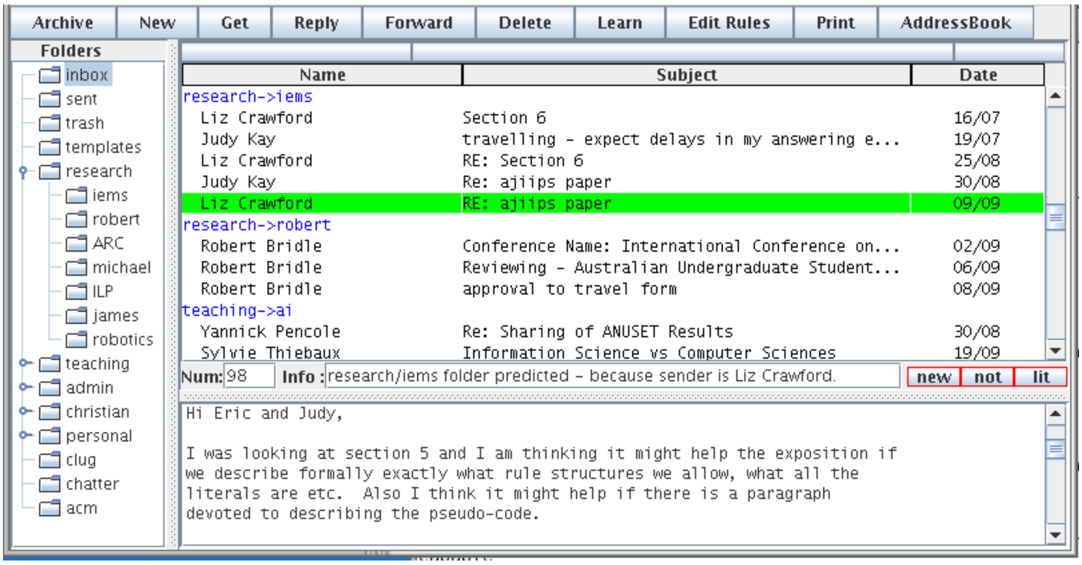


**Figure 22.4.** Partial screen shot from the intelligent email sorting system I-EMS. (Screen shot courtesy of Eric McCreath.)

set of learned rules will sometimes incorrectly predict how the user would classify a message, the interface must be designed in such a way that incorrect predictions will have minimal consequences. New approaches to the general problem continue to appear (see, e.g., Surendran, Platt, & Renshaw, 2005). Two systems that have been fairly widely deployed have been SWIFTFILE (Segal & Kephart, 1999, 2000), which was incorporated into LOTUS NOTES; and POPFILE (available in early 2006 from http://popfile.sourceforge.net/cgi-bin/wiki.pl), a public domain program which is used mainly for spam filtering but which can also learn to sort messages into a limited number of user-specific folders.[2]

Another traditional task in this category is the scheduling of meetings and appointments (Mitchell, Caruana, Freitag, McDermott, & Zabowski, 1994; Maes, 1994; Horvitz, 1999; Gervasio, Moffitt, Pollack, Taylor, & Uribe, 2005): By learning the user's preferences for particular meeting types, locations, and times of day, a system can tentatively perform part of the task of entering appointments in the user's calendar.

The primary benefits of this form of adaptation are savings of time and effort for the user. The potential benefits are greatest where the system can perform the entire task without input from the user. In most cases, however, the user is kept in the loop (as with I-EMS), because the system's ability to predict what the user would want done is limited (cf. the section on usability challenges below).

### Adapting the Interface

A different way of helping a person to use a system more effectively is to adapt the user interface so that it fits better with the user's way of working with the system. Interface elements that have been adapted in this way include menus, icons, and the system's processing of signals from input devices such as keyboards.

An example that will be familiar to most readers is provided by the SMART MENUS feature that has been found in Microsoft operating systems since WINDOWS 2000. Figure 22.6 illustrates the basic mechanism: An infrequently used menu option is initially hidden from view; it appears in the main part of a menu only after the user has selected it for the first time. (It will be removed later if the user does not select it often enough.) The idea is that in the long run the menus should contain just the items that the user has accessed frequently (at least recently), so that the user needs to spend less time searching within menus.

Some informative studies related to SMART MENUS have been conducted by McGrenere and colleagues. In a field study with experienced users of WORD 2000, McGrenere, Baecker, and Booth (2002) compared the SMART MENUS of WORD 2000 with (a) traditional static menus and (b) an alternative approach to reducing the number of functions that confront users: Their variant MSWORD PERSONAL is an *adaptable* system: It provides a reasonably intuitive and convenient way for users to add and remove menu functions. After working with MSWORD PERSONAL for several weeks, most of the users in the study preferred this adaptable system to the normal WORD 2000 with SMART MENUS, and the users who had been classified as "feature-shy" appeared to benefit most. But as is typical in studies like this (as will be discussed below), quite a variety of attitudes about the relative merits of the three approaches to adapting menu content were shown by the subjects. As the authors point out, it seems worthwhile to consider design solutions that com-

[2]For examples of approaches to support for email management that do not involve adaptation to individual users, see, e.g., Gruen et al., 2004; Bälter & Sidner, 2002).

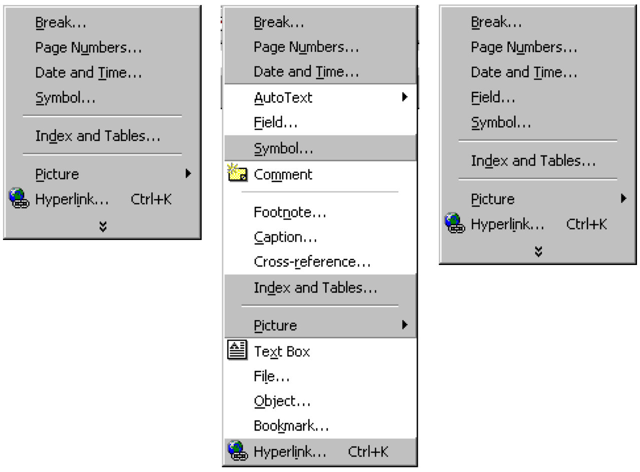


**Figure 22.6.** Example of adaptation in SMART MENUS. (The user accesses the "Insert" menu. Not finding the desired option, the user clicks on the extension arrows and selects the "Field" option. When the user later accesses the same menu, "Field" now appears in the main section.)

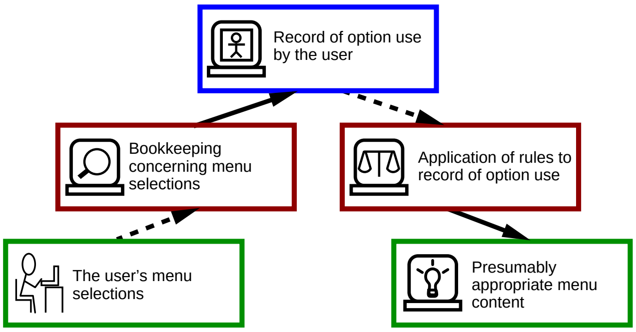


**Figure 22.7.** Overview of adaptation in SMART MENUS.

bine some degree of adaptivity and adaptability. For example, instead of automatically adapting the menus, the system might recommend possible adaptations on the basis of its analysis of the user's menu selections (cf. Bunt, Conati, & McGrenere, 2004).

A more direct experimental comparison by Findlater and McGrenere (2004) involving adaptive menus like SMART MENUS is discussed in the section on empirical methods below.

One promising application of both adaptable and adaptive methods involves taking into account special perceptual or physical impairments of individual users so as to allow them to use a system more efficiently, with minimal errors and frustration (cf. Jacko, Leonard, & Scott, 2008; and Sears, Young, & Feng, 2008). A system in which the two approaches are combined is the WEB ADAPTATION TECHNOLOGY of IBM Research (Hanson & Crayne, 2005), which aims to facilitate web browsing by older adults. With regard to most of the adaptations, such as the reformating of multicolumn text in a single column, the system is adaptable: It provides convenient ways for the user to request the changes. (It would in fact be difficult for a system to determine automatically whether a given user would benefit from one-column formating.) But several changes in the keyboard settings are achieved via automatic adaptation (see Trewin, 2004, for a more detailed discussion). For example, the *key repeat delay* interval is a parameter that determines how long a key (e.g., the left-arrow key) has to be held down before the system starts repeating the associated action (e.g., moving the cursor to the left). Some users require a relatively long key repeat delay because of a tendency to hold keys down relatively long even when they do not want repetition. But asking the user to specify the key repeat delay is not an attractive option: It can be time-consuming to explain what the parameter means; the user herself may have no idea what the best setting is for her; trial and error with different settings can be time-consuming and frustrating; and for some users the optimal setting can change from day to day. The DYNAMIC KEYBOARD component of the WEB ADAPTATION TECHNOLOGY therefore includes an algorithm that analyzes a user's typing behavior to determine an optimal key repeat delay (as well as other parameters); the system then adjusts the parameter in a relatively conservative fashion. Although automatic adjustment of keyboard parameters could under some circumstances make the keyboard unpredictable and hard to use, results obtained in the context of WEB ADAPTATION TECHNOLOGY (Trewin, 2004) revealed no problems of this sort.

### Helping With System Use

Instead of suggesting (or executing) changes to the interface of a given application, a user-adaptive system can adaptively offer information and advice about how to use that application, and perhaps also perform some of the necessary actions itself. There exist various tendencies that make it increasingly difficult for users to attain the desired degree of mastery of the applications that they use. A good deal of research into the development of systems that can take the role of a knowledgeable helper was conducted in the 1980s, especially in connection with the complex operating system UNIX.[3] During the 1990s, such work became less frequent, perhaps partly because of a recognition of the fundamental difficulties involved. In particular, it is often difficult to recognize a user's goal when the user is not performing actions that tend to lead toward that goal. The OFFICE ASSISTANT, an ambitious attempt at adaptive help introduced in MICROSOFT OFFICE 97, was given a mixed reception, partly because of the inherent difficulty of its task but especially because of its widely perceived obtrusiveness (cf. the section on usability challenges below).

Most adaptive help systems to date have been based on the paradigm called *keyhole recognition*: (passively) observing the user and attempting to make useful inferences about her

[3] A collection of papers from this period appeared in a volume edited by Hegner, McKevitt, Norvig, and Wilensky (2001).

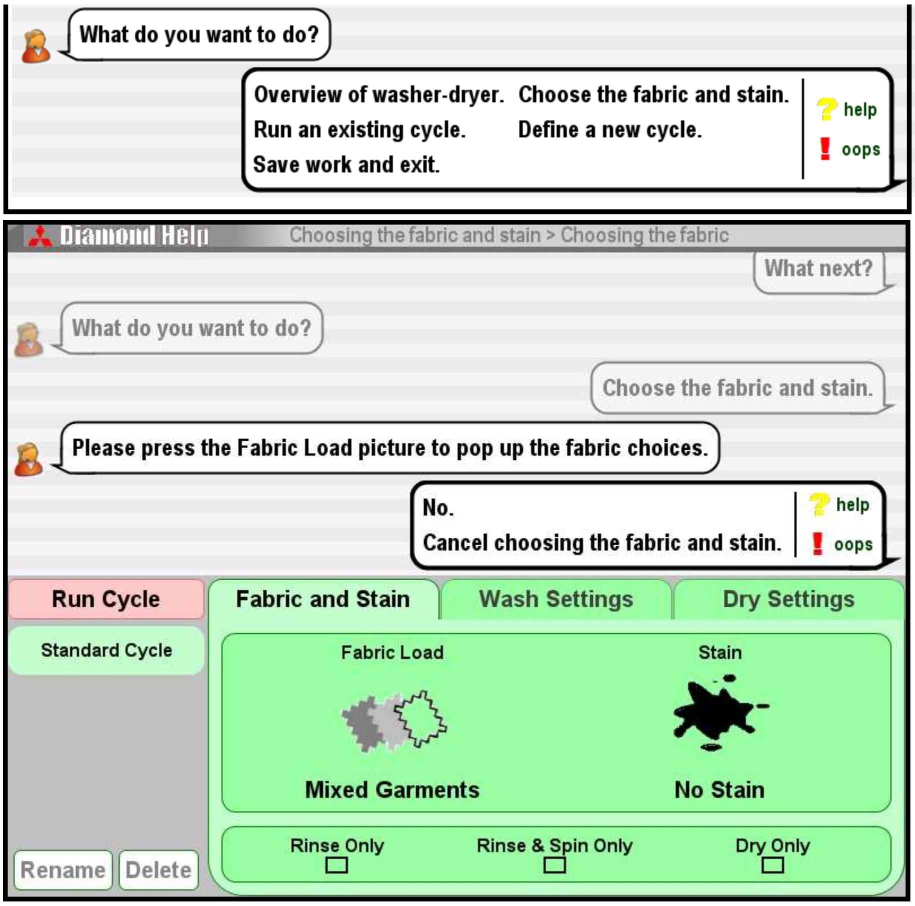


**Figure 22.8.** Example of collaborative assistance offered by DIAMONDHELP. (Explanation in text. Screen shots courtesy of Charles Rich.)

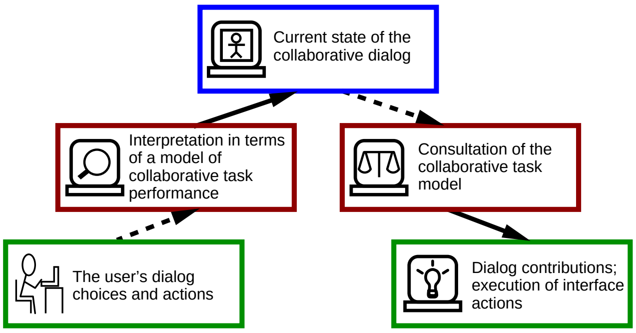


**Figure 22.9.** Overview of adaptation in DIAMONDHELP.

goals and tasks. By contrast, Figure 22.8 shows an example of an alternative approach to intelligent help that has been developed by researchers at Mitsubishi Electric Research Laboratory, which is based on a *collaborative dialog* paradigm (Rich et al., 2005; see Rich & Sidner, 1998, and Rich, Sidner, & Lesh, 2001, for a presentation of the theoretical and technical background). In this demonstration scenario, DIAMONDHELP is collaborating with the user of a feature-rich programmable washer-dryer.[4] Instead of working independently on the problem, the user conducts a dialog with the help system, the goal of the dialog being the execution of the user's task. The dialog contributions of the help system and the user are shown in the "chat" balloons on the left- and right-hand sides, respectively, of the screen. The user's possible dialog contributions are automatically generated from the current collaborative dialog state and offered in a menu inside of his balloon. The user can choose what he wants to say either by touching the appropriate phrase or saying it using speech recognition.[5] For example, in the top part of the figure, the user is offered a choice among three possible top-level tasks, the most complex of which is defining a new

[4]The interface shown in the figures may be displayed on the washer-dryer itself or remotely accessed via a home network.

[5]DIAMONDHELP does not support unrestricted natural language or speech understanding. In a Wizard-of-Oz study (a type of study that will be discussed in the section on empirical methods) involving a prototype help system of this general sort, DeKoven (2004) found that users who were able to employ unrestricted speech would have preferred to have more guidance about what they could say to the system.

cycle. In a typical exchange, the user specifies a goal or subgoal that he would like to achieve, and the system responds by giving instructions and perhaps offering further possible utterances for the user.

This dialog in some ways resembles the interaction with the more familiar type of "wizard" that is often employed for potentially complex tasks such as software installation. The main difference is that the dialogs with DIAMONDHELP can be more flexible, because the system has explicit models of the tasks that the user can perform and is capable of making use of these models in various ways during the dialog. For example, after pressing the Fabric Load picture, the user can continue manipulating the GUI in the lower half of the screen by himself until he requests guidance again, (e.g., by asking "What next?"). Because the user's actions with the interface are reported to the help system, the help system can keep track of how far the user has progressed in the performance of his task. In other words, the help system incorporates a restricted form of the sort of goal and plan recognition that featured prominently in earlier intelligent help systems. In DIAMONDHELP, recognition of the user's actions is relatively likely to be accurate, because of the information that the user has supplied about his goals (see, e.g., Lesh, Rich, & Sidner, 1999). Depending on the experience and the preferences of the user, therefore, the user can rely on the help system to various degrees, ranging from ignoring it, occasionally asking for a hint, or allowing himself to be led step by step through the entire task.

### Mediating Interaction With the Real World

Whereas an intelligent help system aids the user as she uses a complex interactive system, some recently developed systems help the user to cope with the world itself. They do so by acquiring and processing evidence concerning the user's cognitive and/or emotional state and taking actions designed to mitigate any conflict between this state and the demands of the environment.

One common function of systems in this category is to protect people from the flood of incoming messages (via cell phone, instant messaging, email, and other channels) whose number and diversity are increasing with advances in communication technology. When a potential recipient is focusing on some particular task or activity, an adaptive assistant causes messages to be discouraged, delayed, or otherwise buffered until some more appropriate time. One strategy is to provide to the potential initiators of communication information about the state of the recipient. The experimental prototype LILSYS (Figure 22.10) illustrates this strategy. The system continuously updates a user model that contains assessments of its user's availability for communication. The assessments are based on a number of cues that have been found in previous research to be useful predictors of a person's physical presence and/or availability: whether the user (or someone else in the room) is moving or speaking; whether the door is open; whether the user is using the phone or the computer keyboard and mouse; and what events are scheduled in the user's calendar. A hand-crafted model uses this information to arrive at a global assessment of the user's availability; this assessment is in turn displayed to potential communicators.[6] A field study with a small number of users indicated that other persons do in fact adapt their behavior to take into account a LILSYS user's availability, often by changing the nature of their communication rather

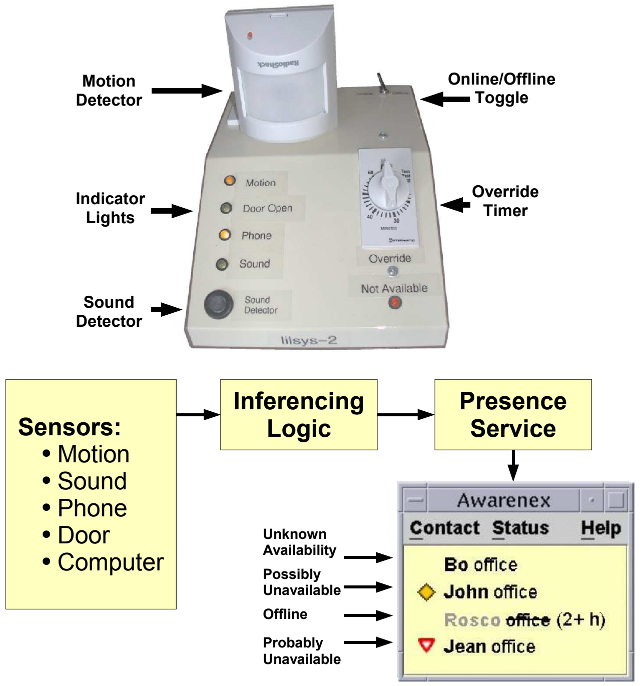


**Figure 22.10.** Above: LILSYS's sensor and data acquisition module; below: the system's data flow and a screen shot of the user interface. (Adapted from Figures 1 and 2 of: "Lilsys: Sensing unavailability," by J. Begole, N. E., Matsakis, & J. C. Tang, 2004, In J. Herbsleb & G. Olson (Eds.), *Proceedings of the 2004 Conference on Computer-Supported Cooperative Work*, pp. 511–514, New York: ACM. Copyright 2004 by ACM. Adapted with permission.)

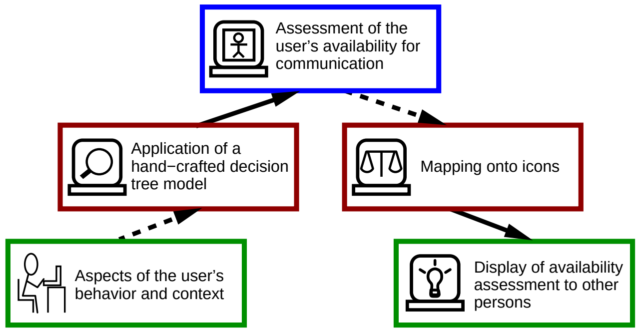


**Figure 22.11.** Overview of adaptation in LILSYS.

[6] In many systems in this category, such as the ones mentioned in the following paragraph, the model for the interpretation of evidence is acquired via machine learning methods on the basis of relevant training data.

than by postponing it. LILSYS users appreciated the possibility of having their availability sensed automatically, as opposed to having to specify it explicitly themselves. For example, they virtually never used the timer switch (visible in Figure 22.10) that allowed them to specify that they were going to be unavailable for a particular period of time. More generally, the modeling of a user's changing cognitive or emotional state appears to be a task for which automatic adaptation is an especially promising approach, simply because people are typically not willing or able to update an explicit self-assessment continually.

In some other availability management systems, decisions about when and how to present messages are made by the system itself on the basis of the user model (see, e.g., Horvitz, Koch, Sarin, Apacible, & Subramani, 2005). A good deal of research has examined effective cues for the recognition of availability and interruptibility (see, e.g., Fogarty et al., 2005; Ho & Intille, 2005; Iqbal, Adamczyk, Zheng, & Bailey, 2005).

A related line of research has focused on the recognition of the mental states of drivers, which is especially important because of safety issues. The modeling of interruptibility is important here as well (see, e.g., Schneider & Kiesler, 2005). But even when no other persons are involved, there are reasons to try to recognize safety-relevant states like drowsiness and stress, so that the system can intervene, for example, by waking the driver up or by playing soothing music. Stress and emotion are manifested in physiological indicators (see, e.g., Healey & Picard, 2000; Lisetti & Nasoz, 2004) and in speech (see, e.g., Fernandez & Picard, 2000; Jones & Jonsson, 2005). Products along these lines have begun to appear in cars, beginning with relatively simple detection methods such as the recognition of long or frequent eye closures. An example of a more complex and comprehensive approach to the modeling of drivers' affective state can be found in the work of Li and Ji (2005).

A general problem with adaptation for the purpose of safety is that the user may come to rely on the adaptation, reducing her own attention to safety.[7] For example, a driver may make less effort to avoid distraction or to remain alert, expecting that the assistant will recognize any dangerous situation and warn him in time. Especially since the recognition of a person's mental states is almost always error-prone, this tendency can eliminate the potential safety benefits of monitoring systems unless appropriate measures are taken (e.g., making warning sounds so unpleasant that the driver will want to avoid relying on them more than necessary).

#### Controlling a Dialog

Much of the early research on user-adaptive systems concerned systems that conducted natural language dialogs with their users (see, e.g., Kobsa & Wahlster, 1989). During the 1990s, attention shifted to interaction modalities that were more widely available and that made it possible in many cases to implement adaptation straightforwardly. Toward the year 2000, advances in the technology of natural language and speech processing (cf. Lai, Karat, & Yankelovich, 2008) led to a recent reawakening of interest in user-adaptive dialog systems (see, e.g., Haller, McRoy, & Kobsa, 1999; Zukerman & Litman, 2001; Litman & Pan, 2002).

Natural language dialog has served as an interaction modality in connection with most of the functions of user-adaptivity discussed in this and the following sections, such as the provision of help and the recommendation of products. But there is also a type of adaptivity which is largely characteristic of natural language dialog: adaptation of the system's *dialog strategy*: a policy for determining when and how the system should provide information, acquire information from the user, and perform other dialog acts.

Adaptation is especially important in spoken dialog systems, such as those that offer information about train departures or flight arrivals via the telephone. Novice users may require extensive explanations and frequent confirmations, but these elements can be unnecessarily time-consuming and frustrating for experienced users. Many deployed systems apply simple adaptation principles that distinguish between new and more experienced users. For example, if a phone-based mail-order system knows that the current caller has previously ordered a product, it may adopt a dialog style that presupposes familiarity with the system. Since it is not always this easy to classify the user, and since asking the user for a self-assessment can be awkward and time-consuming, researchers have looked for ways of adapting to the user on the basis of her behavior during the current dialog (see, e.g., Litman & Pan, 2002). A recent example is given in Figure 22.12, which shows a translation of an example dialog conducted with the KYOTO CITY BUS INFORMATION SYSTEM (Komatani, Ueno, Kawahara, & Okuno, 2005). The system adjusts its assessments of three properties of the caller on the basis of each utterance of the caller: level of skill at conducting dialogs with this system; level of knowledge concerning the domain (i.e., Kyoto and its buses); and desire to complete the dialog quickly ("urgency"). The evidence used is a set of properties such as the amount of information specified in each utterance; whether the user refers to bus stops by their exact names; and whether the user "barges in", interrupting the system before the system has completed an utterance. The dialog excerpt includes several examples that illustrate how the system's dialog behavior is adapted on the basis of the user model. In a controlled experiment, each of 20 subjects conducted dialogs with the system either with the adaptive version or with a version that assumed that all callers had the same properties. Dialogs were completed more quickly with the adaptive variant, and subjects expressed greater satisfaction.[8].

[7] A brief overview of the theory of *risk homeostasis* is given by Wilde (1998), while a recent alternative perspective is offered by Fuller (2005).

[8] See the section on empirical methods for a discussion of the difficulties involved in the interpretation of the results of experiments like this one. Al-

case: [skill level: low] [knowledge level: low] [urgency: low]
Sys: What is your current bus stop, your destination, or specific bus route number?
User: Maruyama Park.
Sys: Let me confirm these in turn.
(← **Explanation is added by [skill level: low].)**
Will you get on at Maruyama Park?
(← **Content is confirmed by [urgency: low].)**
Reply with yes or no.
(← **Instruction is added by [skill level: low].)**
User: Yes.
Sys: Where will you get off the bus?
Specify by bus stop name or a well-known landmark.
(← **Explanation is added by [skill level: low].)**
User: Tofukuji Temple.
Sys: Will you get off the bus at Tofukuji Temple?
Reply with yes or no.
User: Yes.
Sys: Following buses are located.
(← **Explanation is added by [skill level: low].)**
Nearest bus stop to Maruyama Park is Gion.
(← **Domain information is added by [knowledge level: low].)**
Bus number 202 bound for Nishioji-Kujo has departed Higashiyama-Sanjo, which is two stops away.
...

**Figure 22.12.** Excerpt from a dialog with the KYOTO CITY BUS INFORMATION SYSTEM. (Figure 2 of "User modeling in spoken dialogue systems to generate flexible guidance," by K. Komatani, S. Ueno, T. Kawahara, & H. G. Okuno, 2005, *User Modeling and User-Adapted Interaction, 15,* pp. 169–183. Copyright 2005 by Springer Verlag. Reproduced with permission.)

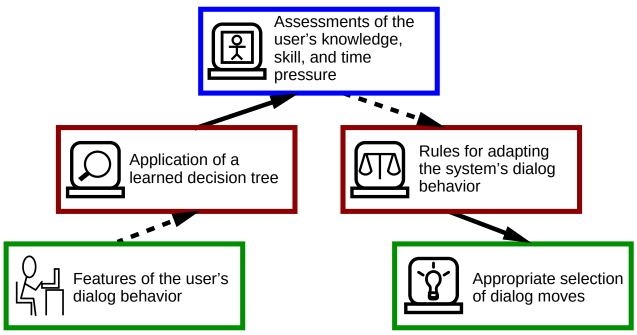


**Figure 22.13.** Overview of adaptation in KYOTO CITY BUS INFORMATION SYSTEM.

Another popular idea about how to adapt dialogs to the user concerns the recognition of negative emotions like anger and frustration, the goal being to transfer callers who express these emotions to human call agents before they are lost as customers. Although there has been a lot of research on the recognition of mental states on the basis of speech in dialogs (see, e.g., Yacoub, Simske, Lin, & Burns, 2003; Liscombe, Riccardi, & Hakkani-Tür, 2005), it remains to be seen how widespread this particular application will become. One possible drawback is that with some systems callers might find it worthwhile to adapt to the adaptation (as with safety-relevant adaptations), feigning emotion in order to get quicker attention.

Other dialog adaptations that are being explored concern stable personal characteristics like gender and age. Since it is possible to recognize these characteristics reasonably well on the basis of speech, a system might adopt a voice or dialog style that designers thought appropriate for the age and/or gender in question (see, e.g., Müller, Wittig, & Baus, 2003).

## FUNCTIONS: SUPPORTING INFORMATION ACQUISITION

We are constantly hearing that information overload is a typical problem of our age, especially because of the explosive growth of the internet and in particular the world-wide web. In addition to the vast number of electronic documents of various sorts, users now have access to a vast number of products available for sale, people that they can get in touch with, and systems that can teach them about some topic. The second major type of function of user-adaptive systems is to help people to find what they need in a form that they can deal with.

### Helping Users to Find Information

We will first look at the broad class of systems that help the user to find relevant electronic documents, which may range from brief news stories to complex multimedia objects.

As an especially clear example, consider the situation of a user who, in the year 2006, has heard that a lot of interesting facts and opinions can be found in blogs (web logs), of which dozens of millions are accessible. She would like to be able to read, each day, the articles in blogs that are of particular interest to her. But how is she to find these? She does not know in advance which blogs are especially likely to produce material of interest to her (as, for example, she could specify a well-known newspaper as a promising source of on-line stories if she were interested in news). She could submit queries to a search engine that indexes blogs; but she cannot in general know in advance what topics of interest to her will be covered by the latest blog articles; and given the low quality and lack of authority of many blogs, she will not be confident of receiving good results on any given topic.

An approach to this problem that relies heavily on adaptation to the individual user (called *personalization* in this context) was available at the time of this writing in the site FINDORY (http://findory.com), which offers access to both blogs and news articles. To the new user visiting the blog section of the site, FINDORY offers a page that shows the first few lines of a number of blog articles on different topics (cf. Figure 22.14). The user can then click to read the articles that interest her most. Each selection causes the system to update its model of the user's interests and adapt the selection of

though the KYOTO CITY BUS INFORMATION SYSTEM is accessible to the public, the adaptive features described here are normally turned off, because they lead to slower processing given the currently available infrastructure.


**Figure 22.14.** A small part of a personalized display of FINDORY (http://findory.com, March 2006). (The icon that appears after a title indicates that the entry has been recommended on the basis of the articles that the user has read previously.)

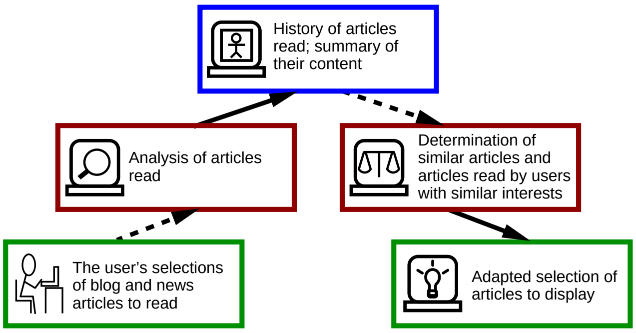


**Figure 22.15.** Overview of adaptation in FINDORY.

blog articles accordingly. For example, if the user chooses an article discussing a copyright infringement suit against a search engine company, further articles concerning copyright issues and search engines are likely to appear, marked with the sunburst icon that is visible in Figure 22.14. If the user clicks on the icon for a recommended article, a page is displayed that explains the recommendation in a style similar to that of AMAZON (cf. Figure 22.1 and the next subsection): with a list of articles that the user has read in the past which are similar to the recommended item in terms of their content (which the system can characterize on the basis of the words in the text) or in terms of the users who have previously read them (Greg Linden, personal communication, February 2006). If the user sees in this way an article that she does not want to be used for recommendations in the future, she can delete it from her *reading history*.

FINDORY's approach relies on the user's being able to identify, early in her use of the system, some articles that interest her and that can therefore serve as examples for the system's learning. This process is facilitated by the opportunities that the user has to issue explicit queries with keywords and to consult trusted sources in the News section of the site. The main advantage of this approach is that the user need not make any effort to specify explicitly what types of content she is interested in. Any such effort would be problematic anyway in that (a) it can be difficult and tedious to describe a large number of general interests accurately, for example by specifying relevant key words; and (b) since interests change over time and as a function of current developments, the user would have to keep updating the descriptions.

More generally speaking, user-adaptive systems that help users find information [9] typically draw from the vast repertoire of techniques for analyzing textual information (and to a lesser extent, information presented in other media) that have been developed in the field of information retrieval. The forms of adaptive support are in part different in three different situations, the first two of which can arise with FINDORY:

*Support for Browsing* In the world-wide web and other hypermedia systems, users often actively search for desired information by examining information items and pursuing cross-references among them. A user-adaptive hypermedia system can help focus the user's browsing activity by recommending or selecting promising items or directions of search on the basis of what the system has been able to infer about the user's information needs. An especially attractive application scenario is that of mobile information access, where browsing through irrelevant pages can be especially time-consuming and expensive. In this context, the best approach may be for the system to omit entirely links that it expects to be less interesting to the individual user. Billsus and Pazzani (2007) describes a case study of an adaptive news server that operated in this way. Stationary systems with greater communication bandwidth tend to include all of the same links that would be presented by a nonadaptive system, highlighting the ones that they consider most likely to be of interest or presenting separate lists of recommended links. As is argued and illustrated by Tsandilas and schraefel (2004), this approach makes it easier for the user to remedy incorrect as-

[9] Surveys of parts of this large area are provided by, among others, Kelly and Teevan (2003) and several chapters in the collection edited by Brusilovsky, Kobsa, and Nejdl (2007).

sessments of the user's interests on the part of the system.

*Support for Query-Based Search or Filtering* Web search engines have been enormously successful and popular in this context, but they have almost always exhibited one limitation: The results presented for a given query have not depended on the interests or previous behavior of the individual user. By contrast, with *personalized search*, the search engine keeps track of user's search history, builds up some sort of model of the user's interests (either by keeping track of and analyzing the user's search history or by asking for an explicit description of interests), and "biases" the results presented accordingly by reordering or filtering the results. A good deal of research (see, e.g., Teevan, Dumais, & Horvitz, 2005; Micarelli, Gasparetti, Sciarrone, & Gauch, 2007) has demonstrated the potential benefits of this strategy. During the year before the writing of this chapter, a personalized variant of the search engine GOOGLE was introduced that sometimes reranked search results on the basis of its record of the user's previous web searching behavior. But it is unclear at the time of this writing how widespread this approach will become. The added value of personalization is less obvious when the user has given an explicit query than when she is simply looking for "something interesting", as is often the case with FINDORY. With an explicit query, it may be feasible and worthwhile for the user to think about an informative description of her interests and to modify her query (perhaps repeatedly) on the basis of the results obtained.

An interesting variant on personalized search is found in the system I-SPY (see, e.g., Smyth et al., 2005): This community-oriented search engine tailors the results of web search queries to an entire community of users, such as the employees of a particular company. It moves upward in the search results list those results that have been clicked on by previous users in the community who had issued the same or similar queries.

*Spontaneous Provision of Information* A number of systems present information that may be useful to the user even while the user is simply working on some task, making no effort to find information. A recent prototype that has been deployed at a research laboratory is the FXPAL BAR (Billsus, Hilbert, & Maynes-Aminzade, 2005). While an employee visits web pages in the course of normal work, the system searches in the background for potentially relevant information (e.g., about company visitors and internal publications). A central design issue for this and similar systems concerns the methods for making the retrieved information available to the user. Presentation of results via means like popup windows risks being obtrusive (cf. the section on usability challenges below), but if the presentation is too subtle, users will often ignore the recommendations and derive little or no benefit from the system. Moreover, the optimal solution in general differs from one user to the next. Billsus et al. (2005) report on studies with a variety of interface solutions for the FXPAL BAR, some of which are adaptable by the user (e.g., the size of a translucent popup window that describes a potentially relevant document).[10]

### Recommending Products

One of the most practically important categories of user-adaptive systems today comprises the product recommenders that are found in many commercial web sites. The best-known such system, the recommender of AMAZON, was discussed briefly in the introduction to this chapter. Looking more closely at Figure 22.1, we can see some distinguishing features of this approach to recommendation. As can be seen from the brief explanations that accompany the recommendations, the system takes as a starting point the information it has about the user's ownership or evaluation of particular products. It then recommends products that are similar in the sense that there is large overlap in the sets of customers that buy them (hence the familiar explanations of the form "Customers who bought this title also bought . . . "). That is, the recommendations are based on a statistical analysis of purchases made by many users, an approach known as *collaborative filtering* (see, e.g., Schafer, Frankowski, Herlocker, & Sen, 2007, for an overview). The products recommended in this way may also happen to be similar in the sense of having the same author or a similar title (as in the examples in the figure), but similarities of this sort can also be conspicuously absent: In the category "Coming soon", the user of Figure 22.1 was recommended the DVD *The Island* because of having positively rated a Sony VAIO notebook PC. As is explained by Linden, Smith, and York (2003), the details of this particular variant of collaborative filtering are due largely to the constraint that it has to be able to cope with AMAZON's millions of products and customers. Although it is generally acknowledged that the recommendations are not always accurate, they can yield notable benefits simply by being better than the generic recommendations (e.g., of top-selling items) that would be presented without personalization.

Some product recommenders allow and require the user to specify her evaluation criteria explicitly, instead of simply rating or purchasing individual items. For example, with the ACTIVE BUYERS GUIDE in Figure 22.16, the user has specified how she intends to use the digital camera that she would like to buy, and the system has recommended three cameras (two of which are visible in this partial screenshot), explaining why each one is suitable in terms of the user's requirements. This *needs-based* approach to recommendation offers a natural alternative to the purely statistical approach of systems like AMAZON when relatively complex and important decisions are involved for which it is worthwhile for the user to think carefully about the attributes of the products in question. It does, however, require that a good deal of knowledge about the features of products and their relationships to user requirements be incorporated in the system.

An intermediate approach between these two extremes is

[10]Influential earlier systems in this category include those of Rhodes (2000) and Budzik, Hammond, and Birnbaum (2001).

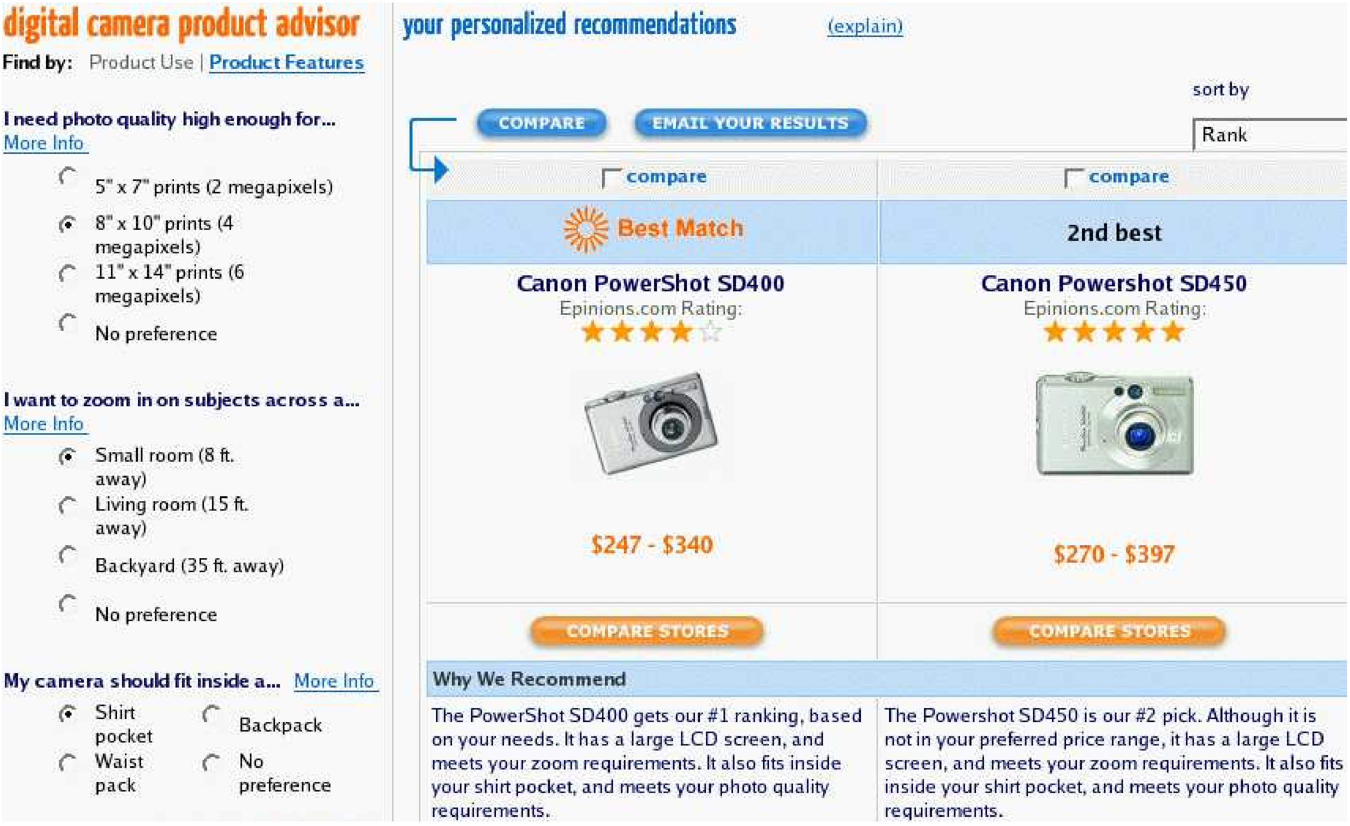


**Figure 22.16.** Partial screen shot from the ACTIVE BUYERS GUIDE recommender for digital cameras. (Screen shot made from http://www.activebuyersguide.com in December 2005.)

the *critiquing* paradigm (see, e.g., Burke, Hammond, & Young, 1997, for an early exposition and McCarthy, Reilly, McGinty, & Smyth, 2005, for an evaluation of some recent advances). The distinguishing feature is an iterative cycle in which the system proposes a product (e.g., a restaurant in a given city), the user criticizes the proposal (e.g., asking for a "more casual" restaurant), and the system proceeds to propose a similar product that takes the critique into account.

Since some products (e.g., movies, vacations) are often used by groups of users, a number of systems have been developed that explicitly address groups (see Jameson & Smyth, 2007, for an overview). The need to address a group rather than an individual has an impact on several aspects of the recommendation process: Users may want to specify their preferences in a collaborative way; there must be some appropriate and fair way of combining the information about the various users' preferences; and the explanations of the recommendations may have to refer to the preferences of the individual group members.

Product recommenders of these various types address several problems that computer users typically experience when they search for products:

1. The user may not know what aspects of the products to attend to or what criteria should determine her decision. Some recommenders either (a) make it less necessary for the user to be explicitly aware of her evaluation criteria (as when collaborative filtering is used) or (b) help the user to learn about her own criteria during the course of the interaction with the system.

2. If the user is unfamiliar with the concepts used to characterize the products, she may be unable to make effective use of any search or selection mechanisms that may be provided. Product recommenders often reduce this communication gap by allowing the user to specify her criteria (if this is necessary at all) in terms that are more natural to her. For example, in Figure 22.16, the user does not need to know in advance how many megapixels she requires in her camera, since the question about photo quality is formulated in everyday terms.

3. Without a recommender, a user might have to read numerous product descriptions in various parts of a website, integrating the information found in order to arrive at a decision. Once a product recommender has acquired an adequate user model, the system can take over a large part of this work, often examining the internal descriptions of a much larger number of products than the user could deal with herself.

From the point of view of the vendors of the products concerned, the most obvious potential benefit is that users will

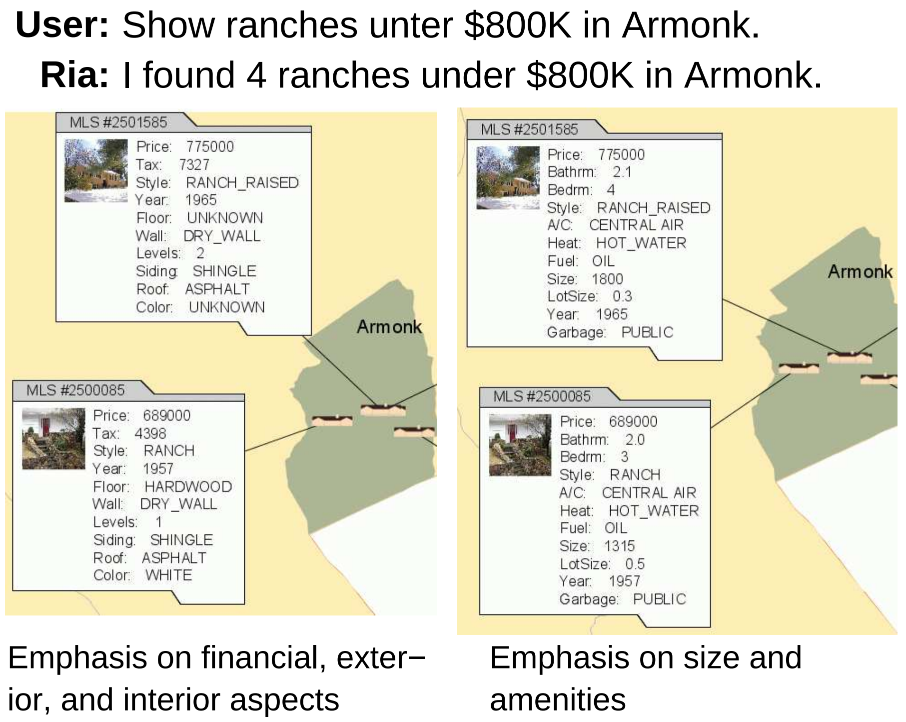


**Figure 22.17.** Two cropped screen shots from the RIA multimedia conversation system. (Screen shots courtesy of Vikram Aggarwal.)

find one or more products that they consider worth buying, instead of joining the notoriously large percentage of browsers who never become buyers. A related benefit is the prospect of cross-selling: the system's model of the user can be employed for the recommendation of further products that the user might not have considered herself. Finally, some vendors aim to build up customer loyalty with recommenders that acquire long-term models of individual customers: If the user believes that the system has acquired an adequate model of her, the user may prefer to use the system again rather than starting from scratch with some other system.

### Tailoring Information Presentation

The ACTIVE BUYERS GUIDE also illustrates a further common function of user-adaptivity: that of presenting information to a user in a way that is especially well suited to that user. In Figure 22.16, the verbal descriptions of the recommended products refer explicitly to the preferences that the user has expressed; if they did not, the user would have to invest more effort to judge how well each product met his requirements.

An instructive example from a research prototype is found in the system RIA (Figure 22.17; Zhou & Aggarwal, 2004), a multimodal system that helps users search for real estate, often presenting information about houses on a map. As the figure shows, the amount of space available for describing a house is limited, so it is important to select the information that is most likely to help the user to decide how to proceed; otherwise, the user may have to request additional information explicitly, which would slow down the interaction. The process of user model acquisition (the three left-most nodes in Figure 22.18) is quite straightforward in RIA: Before the interaction begins, the user is asked a small number of questions about his or her interest and knowledge concerning houses. The sophisticated aspect of the system is the way in which it uses this information—along with information about the houses that satisfy the query—to decide which information to select for presentation and how it is to be ordered. The problem is viewed as one of optimizing the presentation with respect to a large set of constraints.[11] In

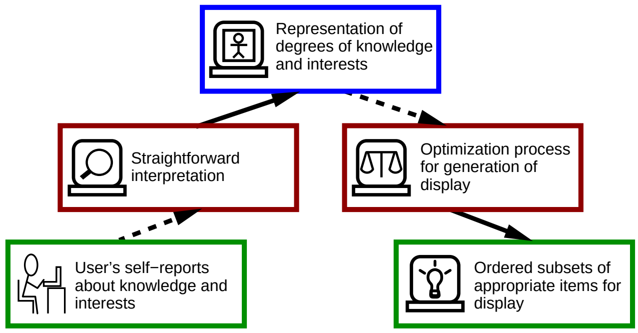


**Figure 22.18.** Overview of adaptation in RIA.

[11] Decisions about what modalities to use for presentation—for example,

an evaluation, the displays generated by this method were found to be similar to those generated by a human designers, who found the task of selecting the appropriate information items to be quite time-consuming (cf. the section on empirical methods for comments on this evaluation method).

Another class of systems in which the tailoring of information to individual users has promise comprises systems that present medical information to patients (see, e.g., Cawsey, Grasso, & Paris, 2007, for an overview).

Properties of users that may be taken into account in the tailoring of documents include: the user's degree of interest in particular topics; the user's knowledge about particular concepts or topics; the user's preference or need for particular forms of information presentation; and the display capabilities of the user's computing device (e.g., web browser vs. cell phone). One strong point of the optimization approach taken with RIA is that all of these factors can be represented and taken into account within a uniform framework.

Even in cases where it is straightforward to determine the relevant properties of the user, the automatic creation of adapted presentations can require sophisticated techniques of natural language generation (see, e.g., Bontcheva & Wilks, 2005) and/or multimedia presentation generation. Various less complex ways of adapting hypermedia documents to individual users have also been developed (see Bunt, Carenini, & Conati, 2007).

### Supporting Collaboration

The increasing tendency for computer users to be linked via networks has made it increasingly feasible for users to collaborate, even in a spontaneous way and without prior acquaintance. A system that has models of a large number of users can facilitate such collaboration by taking into account the ways in which users match or complement each other.

A striking—though not very typical—example is the system AGENTSALON (Sumi & Mase, 2001, 2002), shown in Figure 22.19. The system is used at conferences, in conjunction with a handheld guide (PALMGUIDE) that collects information about exhibits that the user has visited and ratings that she has given of them (the purpose within PALMGUIDE being to make recommendations to the user about other exhibits). When two visitors agree to work with AGENTSALON, the information about them is transferred from their handhelds to AGENTSALON. Like a traditional hostess at a party, AGENTSALON then looks for topics on which the two visitors might be able to hold an interesting conversation—for example, an exhibit about which they gave different ratings. The system tries to get a conversation going by having two animated agents simulate a conversation between these two visitors.

User modeling has been applied in connection with several (partially overlapping) types of collaboration:

text or speech output—are made in a similar way; cf. Zhou, Wen, and Aggarwal (2005).

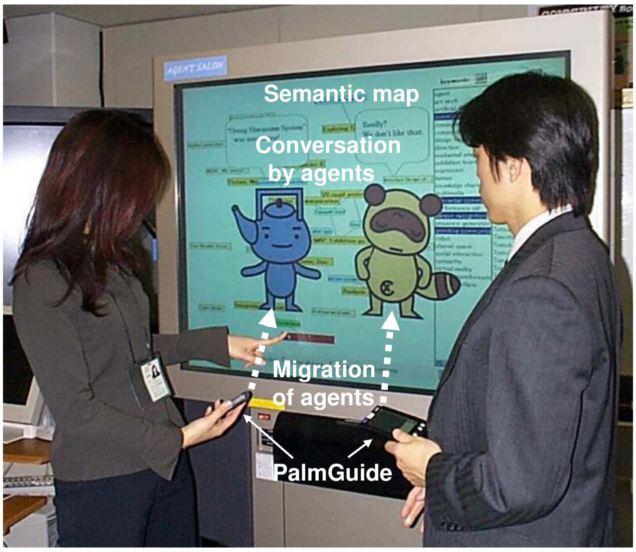


**Figure 22.19.** Attempt by AGENTSALON to stimulate discussion between two conference visitors. (The system has identified an interesting topic by comparing the records of their conference experiences that have been stored on their PDAs. Figure 1 of "AgentSalon: Facilitating face-to-face knowledge exchange through conversations among personal agents," by Y. Sumi & K. Mase, 2001, in *Proceedings of the Fifth International Conference on Autonomous Agents*, pp. 393–400, New York: ACM. Research conducted at ATR Media Information Science Laboratories, Kyoto. Copyright 2001 by the Association for Computing Machinery, Inc. Reproduced with permission.)

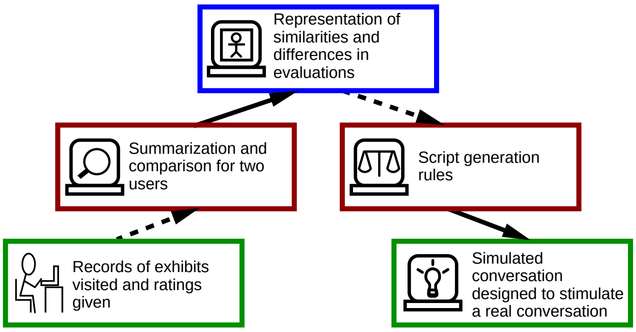


**Figure 22.20.** Overview of adaptation in AGENTSALON.

- In computer-supported learning environments, in which the idea of *collaborative learning* has gained popularity in recent years (see, e.g., Soller, 2007).
- As a way of providing "intelligent help" for complex tasks (see, e.g., Vivacqua & Lieberman, 2000; Aberg & Shahmehri, 2001). Putting a human expert into the loop is a way of avoiding some of the difficulties associated with fully automatic adaptive help systems that were discussed above.
- In environments for computer-supported cooperative

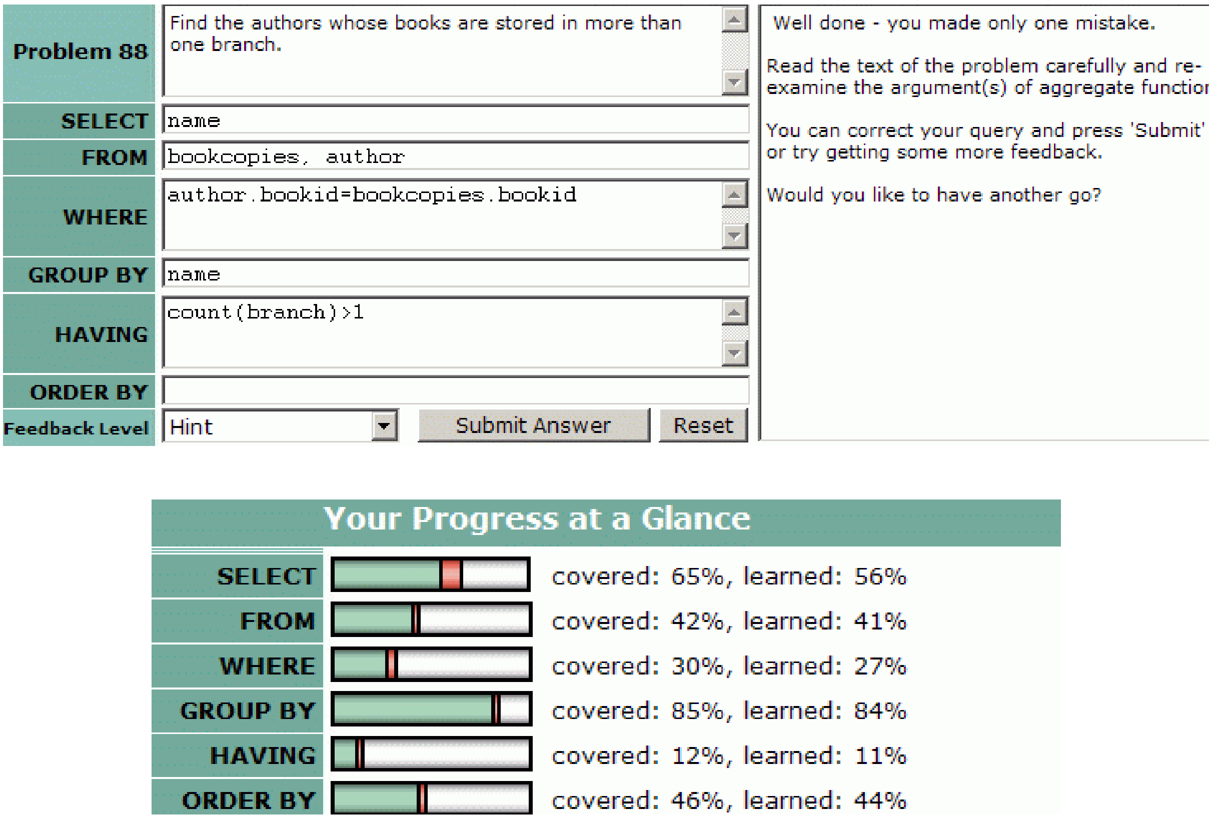


**Figure 22.21.** Screen shots from the SQL-TUTOR. (Above: the main interface; below: display of the learner model. Screen shots courtesy of Antonija Mitrovic.)

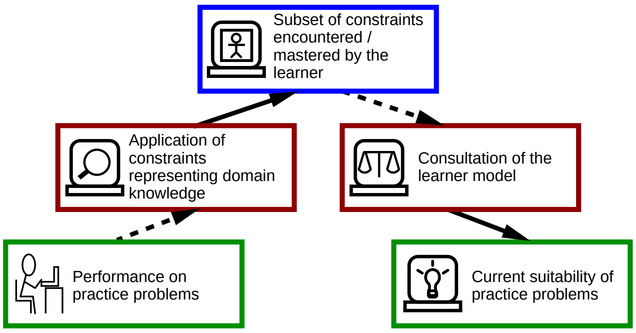


**Figure 22.22.** Overview of adaptation in the SQL Tutor.

work within organizations (see, e.g., Terveen & McDonald, 2005).

### Supporting Learning

Research on *student modeling*—or *learner modeling*, as it has been called more often in recent years—aims to add user-adaptivity to computer-based tutoring systems and learning environments (see, e.g.,Corbett, Koedinger, & Anderson, 1997).[12]

Increasingly, learning environments are being made available on the world-wide web. An example is the SQL-TUTOR (see, e.g., Mitrovic, Suraweera, Martin, & Weerasinghe, 2004), which teaches the database query language SQL.[13] The top part of Figure 22.21 illustrates how the tutor presents a database querying problem and gives feedback on the learner's solution. The lower part of the figure shows a simple visualization of the learner model, which indicates the learner's degree of mastery of each of the six clauses of the SQL SELECT statement. In another part of the interface (not shown in the figure), the system offers suggestions about the type of problem that the learner should attempt next.

A number of different aspects of the SQL-TUTOR have been evaluated in ten studies (see, e.g., Mitrovic et al., 2004; Mitrovic & Ohlsson, 1999), including two studies that showed the value of helping learners to choose the next problem and showing learners their learner model.

Interaction in *intelligent tutoring systems* and *intelligent learning environments* can take many forms, ranging from tightly system-controlled tutoring to largely free exploration

[12]Good sources of literature include the *International Journal of Artificial Intelligence in Education* and the proceedings of the biennial Conferences on Artificial Intelligence in Education (see, e.g., Looi, McCalla, Bredeweg, & Breuker, 2005).

[13]At the time of this writing, the tutor was available to registered students via Addison-Wesley's website DATABASE PLACE (http://www.aw-bc.com/databaseplace/).

by the learner. Aspects of the system that can be adapted to the individual user include: (a) the selection and the form of the instructional information presented; (b) the content of problems and tests; and (c) the content and timing of hints and feedback.

Learner modeling systems may adapt their behavior to any of a broad variety of aspects of the user, such as: (a) the user's knowledge of the domain of instruction, including knowledge acquired prior to and during the use of the system; (b) the user's learning style, motivation, and general way of looking at the domain in question; and (c) the details of the user's current processing of a problem.

The underlying assumption is that the adaptation of the system's behavior to some of these properties of the learner can lead to more effective and/or more enjoyable learning. One series of studies that directly demonstrates the added value of learner-adaptive tutoring is described by Corbett (2001). Many evaluations, however, do not focus on measuring the benefits of adaptivity but rather on comparing alternative variants of the same adaptive system. And in some cases it has been found that the modeling of the learner, however well realized, is not necessary for the effective functioning of the learning environment (see, e.g., VanLehn et al., 2005).

## USABILITY CHALLENGES

Some of the typical properties of user-adaptive systems can lead to usability problems that may outweigh the benefits of adaptation to the individual user. Discussions of these problems have been presented by a number of authors (see, e.g., Norman, 1994; Wexelblat & Maes, 1997; Höök, 2000; Tsandilas & schraefel, 2004; and the references given below in this section). Figure 22.23 gives a high-level summary of many of the relevant ideas, using the metaphor of signs that give warnings and advice to persons who enter a dangerous area.

The Usability Threats shown in the third column concern several generally desirable properties of interactive systems. Those referred to by the top three signs (PREDICTABILITY AND COMPREHENSIBILITY, CONTROLLABILITY, and UNOBTRUSIVENESS) correspond to general usability principles. The remaining two threats, to PRIVACY and to BREADTH OF EXPERIENCE, are especially relevant to user-adaptive systems.

The column Typical Properties lists some frequently encountered (though not always necessary) properties of user-adaptive systems, each of which has the potential of creating particular usability threats.

Each of the remaining two columns shows a different strategy for avoiding or mitigating one or more usability threats: Each of the Preventive Measures aims to ensure that one of the Typical Properties is not present in such a way that it would cause problems. Each of the Remedial Measures aims to ward off one or more threats once it has arisen. The classes of preventive and remedial measures are open-ended, and in fact advances in design and research often take the form of new measures in these classes.

A discussion of all of the relationships indicated in Figure 22.23 would exceed the scope of this chapter, but some remarks will help to clarify the main ideas.

### Threats to Predictability and Comprehensibility

The concept of *predictability* refers to the extent to which a user can predict the effects of her actions. *Comprehensibility* is the extent to which she can understand system actions and/or has a clear picture of how the system works.[14] These goals are grouped together here because they are associated with largely the same set of other variables.

Users can try to predict and understand a system on several different levels of detail.

1. *Exact layout and responses.* Especially detailed predictability is important when interface elements are involved that are accessed frequently by skilled users—for example, icons in control panels or options in menus (cf. the discussion of interface adaptation above). In particular, if the layout and behavior of a system is highly predictable—in fact, essentially identical—over time, skilled users may be able to engage in *automatic processing* (see, e.g., Hammond, 1987): They can use the parts of the interface quickly, accurately, and with little or no attention. In this situation, even minor deviations from complete predictability on a fine-grained level can have the serious consequence of making automatic processing impossible or error-prone.

2. *Success at specific subtasks.* Users may desire only more global predictability and comprehensibility when the system is performing some more or less complex task on the user's behalf (e.g., searching for suitable products on the web): In the extreme case, the system may want only to predict (or evaluate) the quality of the result of a complex system action.

3. *Overall competence.* The most global form of predictability and comprehensibility concerns the user's ability to assess the system's overall level of competence: the degree to which the system tends in general to perform its tasks successfully. With many types of system, high overall competence can be taken for granted; but as we have seen, the processes of acquiring and applying user models do not in general ensure a high degree of accuracy. If the user seriously overestimates the system's competence, she may rely on the system excessively; if she underestimates the system, she may not derive the potential benefits that the system can provide. A factor that is especially important with regard to this global level is the way in which the adaptive part of the system is presented to the user. Some user-adaptive systems, such as AGENTSALON (which was discussed above) and the well-known Microsoft OFFICE ASSISTANT, have employed lifelike characters, for various reasons. As has often been pointed out, such anthropomorphic representations can invoke unrealistically high expectations concerning sys-

[14] The term *transparency* is sometimes used for this concept, but it can be confusing, because it also has a different, incompatible meaning.

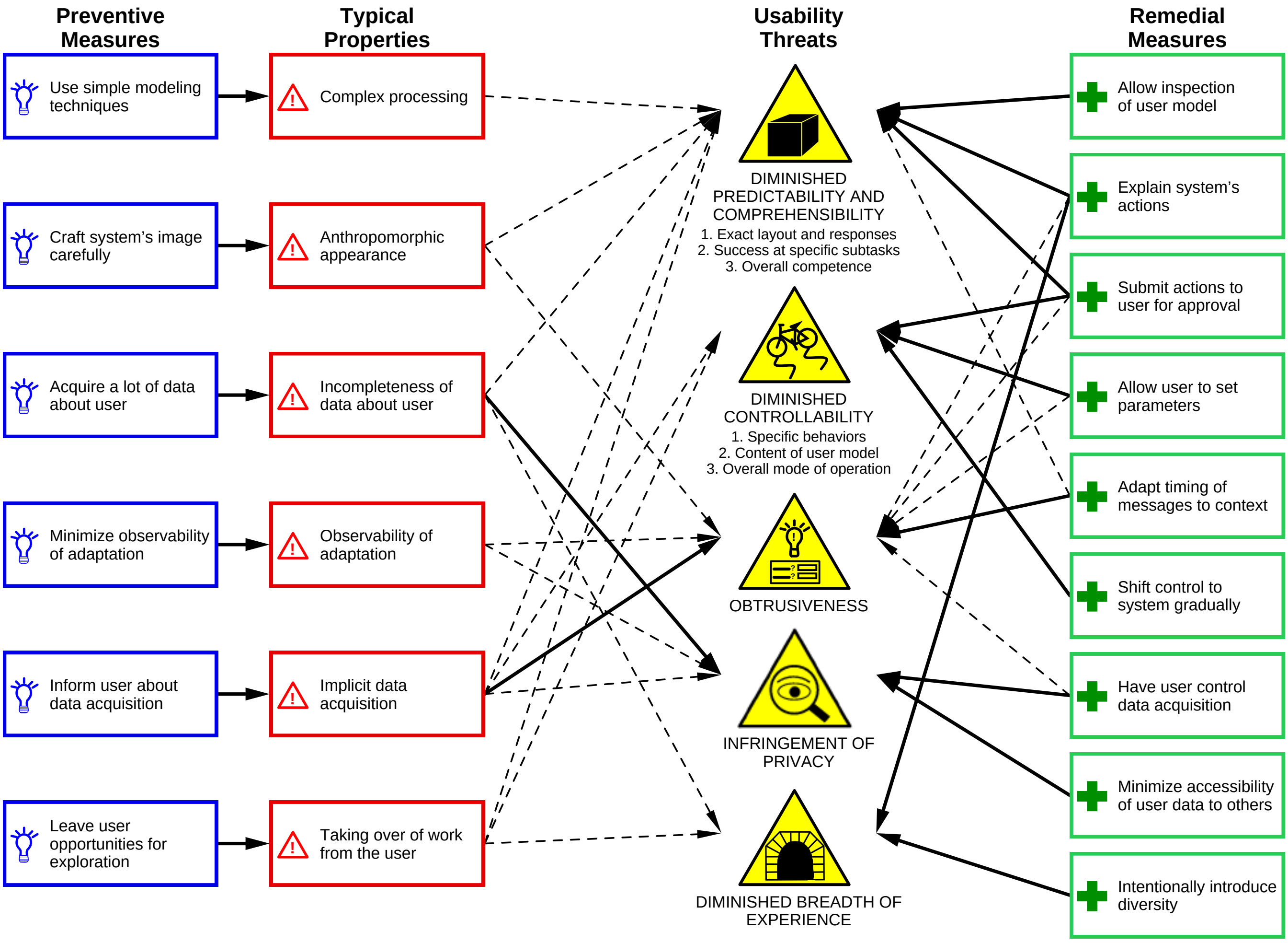


**Figure 22.23.** Overview of usability challenges for user-adaptive systems and of ways of dealing with them. (Dashed arrows denote threats and solid arrows mitigation of threats, respectively; further explanation is given in the text.)

tem competence—not only with regard to capabilities like natural language understanding but also with regard to the system's ability to understand and adapt to the user.

In general, the levels and degrees of predictability and comprehensibility that are necessary or desirable in a given case can depend on many factors, including the function that is being served by the adaptation and the user's level of skill and experience. The same is true of the choice of the strategies that are most appropriate for the achievement of predictability and comprehensibility.

### Threats to Controllability

*Controllability* refers to the extent to which the user can bring about or prevent particular actions or states of the system if she has the goal of doing so. Although controllability tends to be enhanced by comprehensibility and predictability, these properties are not perfectly correlated. For example, when a user clicks on a previously unused option in SMART MENUS, she can predict with certainty that it will be moved to the main part of its menu; but the user has no control over whether this change will be made.

A typical measure for ensuring some degree of control is to have the system submit any action with significant consequences to the user for approval. This measure can cause a threat of *obtrusiveness* (see below); so it is an important interface design challenge to find ways of making recommendations in an unobtrusive fashion that still makes it easy for the user to notice and follow up on them (cf. the earlier discussion of FXPAL BAR).

Like predictability and comprehensibility, controllability can be achieved on various levels of granularity. Especially since the enhancement of controllability can come at a price, it is important to consider what kinds of control will really be desired. For example, there may be little point in submitting individual actions to the user for approval if the user lacks the knowledge or interest required to make the decisions. Wexelblat and Maes (1997) recommend making available several alternative types of control for users to choose from.

### Obtrusiveness

We will use the term *obtrusiveness* to refer to the extent to which the system places demands on the user's attention which reduce the user's ability to concentrate on her primary tasks. This term—and the related words *distracting* and *irritating*—are often heard in connection with user-adaptive systems. Figure 22.23 shows that (a) there are several different reasons why user-adaptive systems can easily turn out to be obtrusive and (b) there are equally many corresponding strategies for minimizing obtrusiveness. Some of these measures can lead straightforwardly to significant improvements—for example, when it is recognized that distracting lifelike behaviors of an animated character are not really a necessary part of the system.

### Threats to Privacy

User-adaptive systems typically (a) gather data about individual users and (b) use these data to make decisions that may have more or less serious consequences. Users may accordingly become concerned about the possibility that their data will be put to inappropriate use. Privacy concerns tend to be especially acute in e-commerce contexts (see, e.g., Cranor, 2004), and with some forms of support for collaboration (see, e.g., Terveen & McDonald, 2005), because in these cases (a) data about the user are typically stored on computers other than the user's own; (b) the data often include personally identifying information; and (c) there may be strong incentives to use the data in ways that are not dictated by the user's own interests. As will be discussed in the next section, different means of acquiring information about users can have different consequences with regard to privacy. On the other hand, many of the measures that can be taken to protect privacy—for example, a policy of storing as little personally identifying data as possible—are not specific to user-adaptive systems (see, e.g., Karat, Karat, & Brodie, 2008).

### Breadth of Experience

When a user-adaptive system helps the user with some form of information acquisition (cf. the second major section of this chapter), much of the work of examining the individual documents, products, and/or people involved is typically taken over by the system. A consequence can be that the user ends up learning less about the domain in question than she would with a nonadaptive system (cf. Lanier, 1995). For example, if the AMAZON visitor for whom recommendations are shown in Figure 22.1 relies heavily on such recommendations (as opposed to browsing freely), he is likely to learn a lot about the books of Frederick Forsyth and about closely related products but little about the full range of books and other media that are available. One point of view here (see, e.g., the remarks of Maes in Shneiderman & Maes, 1997, p. 53) is that it should be up to the user to decide whether she prefers to learn about a given domain or to save time by delegating work to a system. It may be worthwhile to of-

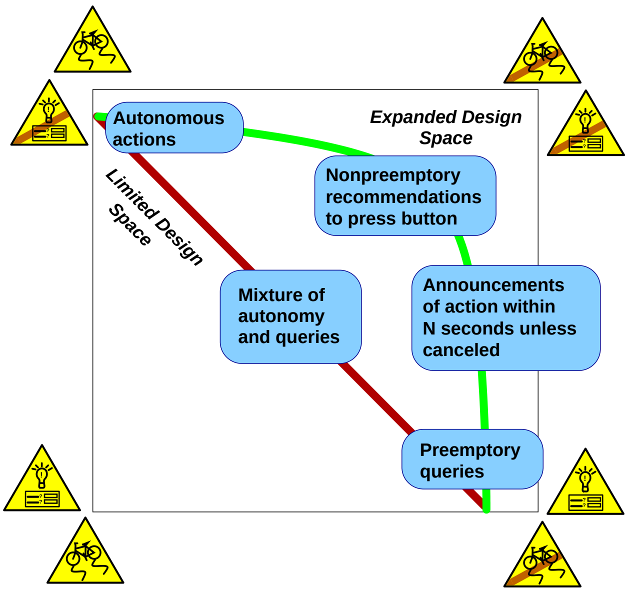


**Figure 22.24.** Illustration of strategies for dealing with tradeoffs among usability goals in user-adaptive systems. (Explanation in text.)

fer the user a continuous spectrum of possibilities between complete control over a task and complete delegation of it. For example, many product recommendation systems, such as AMAZON's, allow users to alternate freely between pursuing the system's recommendations and browsing through product descriptions in the normal way.

Reduction of breadth of experience is especially likely if the system relies more heavily than is necessary on an incomplete user model. The user of Figure 22.1 had (understandably) informed the system about only a tiny proportion of the books that he had ever read and liked; and since the system tends to recommend—and offer a chance to rate—similar items, the system may never obtain much evidence about all of the user's other literary interests. Some systems mitigate this problem by systematically proposing solutions that are *not* dictated by the current user model (see, e.g., Ziegler, McNee, Konstan, & Lausen, 2005, for a method that is directly applicable to recommendation lists such as AMAZON's; and Linden, Hanks, & Lesh, 1997, and Shearin & Lieberman, 2001, for methods realized in different types of recommenders).

### Dealing With Tradeoffs

As can be seen in Figure 22.23, the designer who attempts to combat a particular usability threat will often have to deal with a threat to some other usability goal. The most obvious tradeoffs involve UNOBTRUSIVENESS. In particular, steps taken to enhance control or to protect privacy often require the user to perform additional actions, input additional information, and/or pay attention to additional system messages. Dealing with tradeoffs of this sort is complicated by the fact that users often differ markedly in the relative priority that they assign to each of the conflicting goals.

Figure 22.24 illustrates some general points, referring for concreteness to a recently developed prototype OFFICE CONTROL SYSTEM (Cheverst et al., 2005).This system first observes how the occupant of an office tends to operate various devices such as the fan and the window shades; it then tries to help the user by performing some actions autonomously (e.g., opening the window when particular weather conditions prevail and there is no visitor in the office). An early version offered two ways of dividing the work between the user and the system for each type of action: The system could either perform actions of that type autonomously or request the user's permission with a preemptory dialog box on the user's normal computer screen. Different users chose the latter option for different proportions of the available actions, reflecting different priorities for the goals of unobtrusiveness and controllability, respectively. In terms of the tradeoff graph shown in Figure 22.24, users chose different points on the straight diagonal line. But despite this freedom of choice, users were often not satisfied with the overall usability of the prototype. A significant improvement in acceptance was achieved when the designers *expanded the design space*: They introduced a separate small screen for the OFFICE CONTROL SYSTEM, in which information and requests for confirmation could be offered in several ways that are not very familiar in everyday graphical user interfaces (though they are familiar in industrial and traffic contexts; see, e.g., Wickens & Hollands, 2000). As Figure 22.24 indicates, these additional forms of interaction represented a more favorable combination of degrees of unobtrusiveness and controllability at least for some of the users some of the time.

Consistent with more complex examples (see, e.g., Jameson & Schwarzkopf, 2002; Billsus & Pazzani, 2007), this small case study illustrates two general points: 1. When dealing with tradeoffs among the usability goals discussed here, it can be important to offer alternative solutions for users with different priorities. 2. It may be equally important to consider relatively novel interface design solutions that may spare users the need to choose among unsatisfactory alternatives.

### OBTAINING INFORMATION ABOUT USERS

Some of the usability challenges discussed in the previous section are closely connected with the ways in which information about individual users is acquired—a factor which also largely determines the success of a system's adaptation. The next two subsections will look, respectively, at (a) information that the user supplies to the system explicitly for the purpose of allowing the system to adapt; and (b) information that the system obtains in some other way.

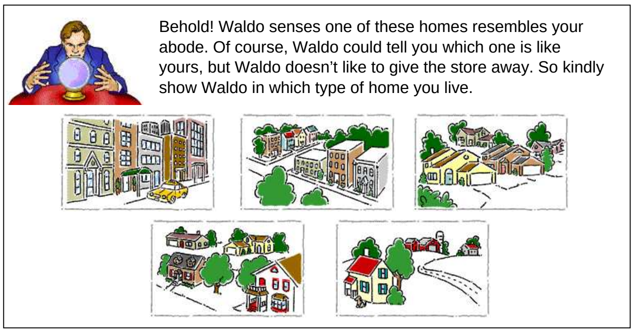


**Figure 22.25.** Example of a screen with which the LIFESTYLE FINDER elicits demographic information. (Figure 3 of "Lifestyle Finder: Intelligent user profiling using large-scale demographic data," by B. Krulwich, 1997, *AI Magazine, 18*(2), pp. 37–45. Research conducted at the Center for Strategic Technology Research of Andersen Consulting (now Accenture Technology Labs). Copyright 1997 by the American Association for Artificial Intelligence. Adapted with permission.)

#### Explicit Self-Reports and -Assessments

##### *Self-Reports About Objective Personal Characteristics*

Information about objective properties of the user (such as age, profession, and place of residence) sometimes has implications that are relevant for system adaptation—for example, concerning the topics that the user is likely to be knowledgeable about or interested in. This type of information also has the advantage of changing relatively infrequently. Some user-adaptive systems request information of this type from users, but the following caveats apply:

1. Specifying information such as profession and place of residence may require a fair amount of tedious menu selection and/or typing.

2. Since information of this sort can often be used to determine the user's identity, a user may justifiably be concerned about privacy issues. Even in cases where such concerns are unfounded, they may discourage the user from entering the requested information.

A general approach is to (a) restrict requests for personal data to the few pieces of information (if any) that the system really requires; and (b) explain the uses to which the data will be put. A number of suggestions about how the use of personally identifying data can be minimized are given by Cranor (2004). An especially creative approach was tried in the web-based LIFESTYLE FINDER prototype (Figure 22.25; Krulwich, 1997), which was characterized by a playful style and an absence of requests for personally identifying information. Of the users surveyed, 93% agreed that the LIFESTYLE FINDER's questions did not invade their privacy.

It is sometimes possible to avoid requests for explicit input about personal characteristics by accessing sources where similar information has already been stored (a strategy that will be discussed in the next subsection).

#### *Self-Assessments of Interests and Knowledge*

It is sometimes helpful for a user-adaptive system to have an assessment of a property of the user that can be expressed naturally as a position on a particular general dimension: the level of the user's interest in a particular topic, the level of her knowledge about it, or the importance that the user attaches to a particular evaluation criterion. Often an assessment is arrived at through inference on the basis of indirect evidence, as with the assessments of a learner's knowledge in the SQL-TUTOR (Figure 22.21). But it may be necessary or more efficient to ask the user for an explicit assessment. For example, it would be difficult for the ACTIVE BUYERS GUIDE recommender shown in Figure 22.16 to estimate the importance that the user attaches to photo quality without asking the user directly. The scales in this figure illustrate good practice in that they make clear the meaning of the various possible answers, instead of asking "How important is photo quality to you (on a scale from 1 to 5)?"[15]

Because of the effort involved in this type of self-assessment, it is in general worthwhile to consider ways of minimizing such requests, making responses optional, ensuring that the purpose is clear, and integrating the self-assessment process into the user's main task (see, e.g., Tsandilas & schraefel, 2004, for some innovative ideas about how to achieve these goals).

#### *Self-Reports on Specific Evaluations*

Instead of asking the user to describe her interests explicitly, some systems try to infer the user's position on the basis of her explicitly evaluative responses to specific items. AMAZON's rating scales (Figure 22.1) illustrate one form that this type of input can take; other forms include checkboxes and icons (e.g., "thumbs-up" and "thumbs-down"). The items that the user evaluates can be (a) items that the user is currently experiencing directly (e.g., the current web page); (b) actions that the system has just performed, which the user may want to encourage or discourage (see, e.g., Wolfman, Lau, Domingos, & Weld, 2001); (c) items that the user must judge on the basis of a description (e.g., the abstract of a talk; a table listing the attributes of a physical product); or (d) the mere name of an item (e.g., a movie) that the user may have had some experience with in the past (see, e.g., Figure 22.1). The cognitive effort required depends in part on how directly available the item is: In the third and fourth cases just listed, the user may need to perform memory retrieval and/or inference in order to arrive at an evaluation.

Even when the effort is minimal, users often do not like to bother with explicit evaluations that do not constitute a necessary part of the task they are performing. For this reason, many designers try to get by with the type of nonexplicit input discussed in the following section. For example, FINDORY (Figure 22.14) could allow the user to rate the various news stories and blogs presented, but instead it just interprets the user's behavior in selecting items to read.

#### *Responses to Test Items*

In systems that support learning, it is often natural to administer tests of knowledge or skill. In addition to serving their normal educational functions, these tests can yield valuable information for the system's adaptation to the user. An advantage of tests is that they can be constructed, administered, and interpreted with the help of a large body of theory, methodology, and practical experience (see, e.g., Wainer, 2000).

Outside of a learning context, users are likely to hesitate to invest time in tests of knowledge or skill unless these can be presented in an enjoyable form (see, e.g., the color discrimination test used by Gutkauf, Thies, & Domik, 1997, to identify perceptual limitations relevant to the automatic generation of graphs). Trewin (2004) reports on experience with a brief typing test that was designed to identify helpful keyboard adaptations: Some users who turned out to require no adaptations were disappointed that their investment in the test had yielded no benefit. As a result, Trewin decided that adaptations should be based on the users' naturally occurring typing behavior.

### Nonexplicit Input

The previous subsection has given some examples of why designers often look for ways of obtaining information about the user that does not require any explicit input by the user.

#### *Naturally Occurring Actions*

The broadest and most important category of information of this type includes all of the actions that the user performs with the system that do not have the purpose of revealing information about the user to the system. These actions may range from major actions like purchasing an expensive product to minor ones like scrolling down a web page. The more significant actions tend to be specific to the particular type of system that is involved (e.g., e-commerce sites vs. learning environments). Within some domains, there has been considerable research on ways of interpreting particular types of naturally occurring user actions. For example, researchers interested in adaptive hypertext navigation support have developed a variety of ways of analyzing a user's navigation actions to infer the user's interests and/or to propose navigation shortcuts (see, e.g., Mobasher, 2007).

In their purest form, naturally occurring actions require no additional investment by the user, because they are actions that the user would perform anyway. The main limitation is that they are hard to interpret; for example, the fact that a given web page has been displayed in the user's browser for 4 minutes does not reveal with certainty which (if any) of the text displayed on that page the user has actually read. Some designers have tried to deal with this tradeoff by designing the user interface in such a way that the naturally

[15] Some further guidance concerning the formulation of questions of this general sort is given, for example, by Ozok (2008).

occurring actions are especially easy to interpret. For example, a web-based system might display just one news story on each page, even if displaying several stories on each page would normally be more desirable.

The interpretation of naturally occurring actions by the system can raise privacy and comprehensibility issues (cf. Figure 22.23) that do not arise in the same way with explicit self-reports and self-assessments of the types discussed earlier in this section: Whereas the latter way of obtaining information about the user can be compared with interviewing, the former way is more like eavesdropping—unless the user is informed about the nature of the data that are being collected and the ways in which they will be used (cf. Cranor, 2004).

#### *Previously Stored Information*

Sometimes a system can access relevant information about a user which has been acquired and stored independently of the system's interaction with the user:

1. If the user has some relationship (e.g., patient, customer) with the organization that operates the system, this organization may have information about the user that it has stored for reasons unrelated to any adaptation, such as the user's medical record (see Cawsey et al., 2007, for examples) or address.

2. Relevant information about the user may be stored in publicly available sources such as electronic directories or web homepages. For example, Pazzani (1999) explores the idea of using a user's web homepage as a source of information for a restaurant recommending system.

3. If there is some other system that has already built up a model of the user, the system may be able to access the results of that modeling effort and try to apply them to its own modeling task. There is a line of research that deals with *user modeling servers* (see, e.g., Kobsa, 2007): systems that store information about users centrally and supply such information to a number of different applications. Some of the major commercial personalization software is based on this conception.

Relative to all of the other types of information about users, previously stored information has the advantage that it can in principle be applied right from the start of the first interaction of a given user with a given system. On the other hand, the interpretability and usefulness of the information in the context of the current application may be limited. Moreover, questions concerning privacy and comprehensibility may be even more important than with the interpretation of naturally occurring actions.

#### *Low-Level Indices of Psychological States*

The next two categories of information about the user have become practically feasible only in recent years, with advances in the miniaturization of sensing devices.

The first category of sensor-based information (discussed at length in the classic book of Picard, 1997) comprises data that reflect aspects of a user's psychological state. Some of the application scenarios in which this type of information can be useful were discussed in the section on systems that mediate interaction with the real world.

Two categories of sensing devices have been employed: (a) devices attached to the user's body (or to the computing device itself) that transmit physiological data, such as electromyogram signals, the galvanic skin response, blood volume pressure, and the pattern of respiration (see Lisetti & Nasoz, 2004, for an overview); and (b) video cameras and microphones that transmit psychologically relevant information about the user, such as features of her facial expressions (see, e.g., Bartlett, Littlewort, Fasel, & Movellan, 2003) or her speech (see, e.g., Liscombe et al., 2005).

With both categories of sensors, the extraction of meaningful features from the low-level data stream requires the application of pattern recognition techniques. These typically make use of the results of machine learning studies in which relationships between low-level data and meaningful features have been learned.

One advantage of sensors is that they supply a continuous stream of data, the cost to the user being limited to the physical and social discomfort that may be associated with the carrying or wearing of the devices. These factors are still significant now, but further advances in miniaturization—and perhaps changing attitudes as well—seem likely to reduce their importance.

#### *Signals Concerning the Current Surroundings*

As computing devices become more portable, it is becoming increasingly important for a user-adaptive system to have information about the user's current surroundings. Here again, two broad categories of input devices can be distinguished (see Krüger, Baus, Heckmann, Kruppa, & Wasinger, 2007, for a discussion of a number of specific types of devices).

1. Devices that receive explicit signals about the user's surroundings from specialized transmitters. Some mobile systems that are used outdoors employ GPS (Global Positioning System) technology. More specialized transmitters and receivers are required, for example, if a portable museum guide system is to be able to determine which exhibit the user is looking at.

2. More general sensing or input devices. For example, Schiele, Starner, Rhodes, Clarkson, and Pentland (2001) describe the use of a miniature video camera and microphone (each roughly the size of a coin) that enable a wearable computer to discriminate among different types of surroundings (e.g., a supermarket vs. a street). The use of general-purpose sensors eliminates the dependence on specialized transmitters. On the other hand, the interpretation of the signals requires the use of sophisticated machine learning and pattern recognition techniques.

## SPECIAL CONSIDERATIONS CONCERNING EMPIRICAL METHODS

The full repertoire of empirical methods in human-computer interaction is in principle applicable to user-adaptive systems. This section will focus on some methods that are more important for user-adaptive systems than for other types and on some typical problems that need to be dealt with. But this focused discussion should not obscure the fact that a lot of empirical work with user-adaptive systems looks the same as with other systems.[16]

### Use of Data Collected With a Nonadaptive System

The key difference between user-adaptive systems and other interactive systems is the inclusion of some method for acquiring and exploiting a user model. This feature gives rise to a type of empirical study that is largely unique to user-adaptive systems: studies in which the accuracy of the modeling methods is evaluated.

This type of evaluation can often be performed even if there exist no user-adaptive systems that employ the user modeling method in question. What is needed are (a) some implementation of the adaptation algorithm, not necessarily embedded in any interactive system; and (b) a database of behavioral data from a number of users who have used a relevant *nonadaptive* system. The researcher can then apply the modeling method to the data in order to determine how well the system would adapt to the users in question.

A number of studies of this type were conducted with the I-EMS system (see McCreath et al., 2005, and the discussion earlier in this chapter). In one case, the researchers wanted to find out whether a user who had defined a number of hand-crafted email sorting rules could benefit from having automatically learned rules applied to messages that were not covered by the hand-crafted rules. One simulation was performed on 5100 email messages that had been sorted by a single user within a nonadaptive email client over a 3-month period. In the order in which the messages had been received, they were presented to one of the system's learning algorithms in batches of 100, along with information about how the user had sorted them, so that the system could continually refine its set of learned rules. After each batch of 100 messages, the accuracy of the updated set of rules was evaluated: The system was asked to make predictions about the next batch of messages before being told how the user had in fact sorted them, and these predictions were compared with the user's actual behavior. Several indices of the system's performance were computed, one of which is shown in Figure 22.26: the percentage of messages for which the learned rules made no prediction (i.e., where the appropriate folder was "unknown" to the system). The middle curve in the graph shows how this percentage gradually decreased over time. The uppermost curve shows the corresponding results for the case where only the user's hand-crafted rules were used for prediction; the lowest curve shows the results where both types of rules were applied, with the hand-crafted rules taking precedence in the case of disagreement. It can be seen that the joint use of both sets of rules gave the best results in terms of avoiding "unknown" predictions. Since according to other indices the accuracy of the combined set of rules was as good as that of the hand-crafted rules alone, the system's performance was best overall with the combined set.

Note that it was not necessary, for this evaluation, to create three different versions of I-EMS and have them used for months. In addition to being time-consuming, this procedure would allow less direct accuracy comparisons. By contrast, using existing email corpora, McCreath et al. (2005) were able to perform simulations that shed light on many properties of the algorithms used.

The appeal of this type of evaluation, in terms of being able to yield numerous interpretable results with minimal involvement of actual users, is so great that researchers sometimes seem to lose sight of the fact that studies with real users are likewise essential. No simulation study, for example, can reveal how well the design of the I-EMS interface shown in Figure 22.4 corresponds with the way users like to deal with incoming email, or how helpful users find the explanations that the system offers for its predictions.

### Early Studies of Usage Scenarios and User Requirements

In the field of human-computer interaction as a whole, it is expected that user-centered design should begin with a study of contexts of use, usage scenarios, properties of users, and user requirements. To date, this strategy has been applied less frequently in the design of user-adaptive systems—perhaps because the designers less frequently come from an HCI background, often specializing instead in the development of adaptive algorithms. Early user studies are actually at least as important with novel user-adaptive systems as with other types of system: It is often not clear in advance whether adaptation will yield added value and achieve acceptance in a particular context. Careful attention to the requirements and contexts of users may greatly increase the likelihood of success—or at least warn the designers at an early stage if a particular usage scenario is not promising for the sort of adaptive interaction that they envision.

A positive example of early attention to user requirements is found in the development of the museum guide HYPERAUDIO, which was developed as a prototype in the 1990s (see, e.g., Petrelli & Not, 2005, for a retrospective discussion). Studying the attitudes and behavior of museum visitors at an early stage in the system's design, the researchers found that many visitors enjoy guided tours but that few visitors want to spend time interacting with technical devices. These two findings, along with others, led to a modification of the original conception of HYPERAUDIO: They suggested the appropriateness of a museum guide that selects information for

[16] More extended discussions of empirical methods for user-adaptive systems are provided by Gena and Weibelzahl (2007); Höök (2000); and Langley and Fehling (1998).

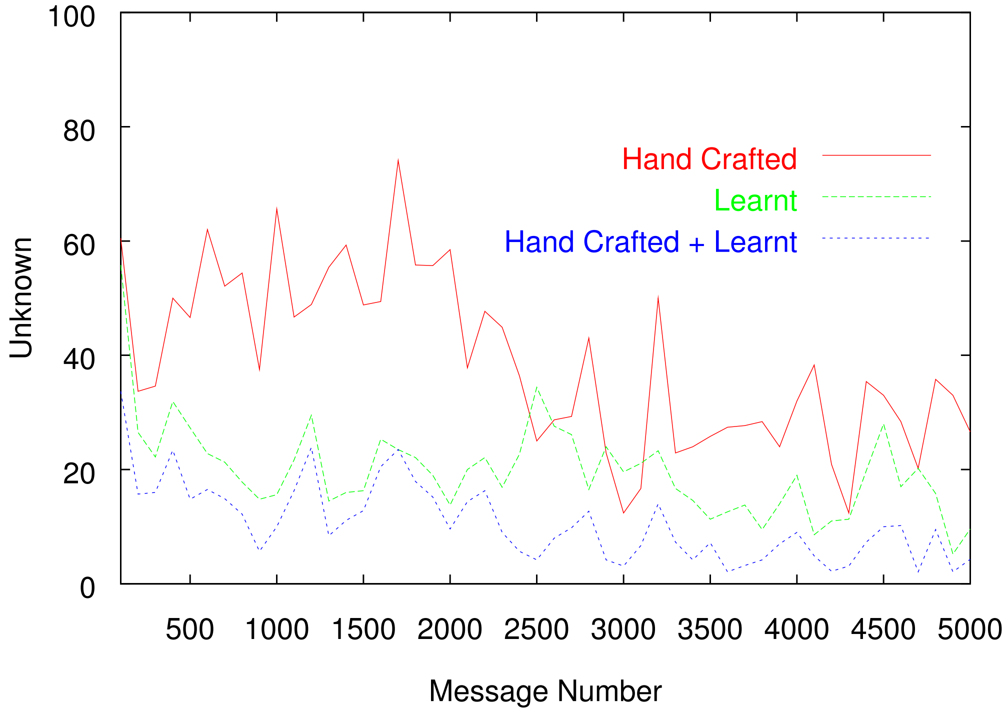


**Figure 22.26.** Results of one of the simulation experiments performed with I-EMS's algorithms. (Explanation in text. Figure courtesy of Eric McCreath.)

presentation on the basis of the user's behavior and location, requiring little or no explicit input.

#### Wizard-of-Oz Studies

Systems that adapt to their users are in one methodological respect similar to systems that make use of speech (cf. Lai et al., 2008): They attempt to realize a capability that is so far, at least in many contexts, possessed to the highest degree by humans. Consequently, as with speech interfaces, valuable information can sometimes be obtained from a *Wizard-of-Oz study*: In a specially created setting, a human takes over a part of the processing of the to-be-developed system for which humans are especially well suited (cf. Lai et al., 2008; Beaudouin-Lafon & Mackay, 2008).

One example is the Wizard-of-Oz study that was conducted early in the development of the LUMIÈRE intelligent help system (Horvitz, Breese, Heckerman, Hovel, & Rommelse, 1998), which formed the basis for the OFFICE ASSISTANT, which was introduced in Microsoft OFFICE 97. In this study, subjects working with a spreadsheet were told that an experimental help system would track their activity and make guesses about how to help them. They received the advice via a computer monitor. The advice was actually provided by usability experts who, working in a separate room, viewed the subjects' activity via a monitor and conveyed their advice by typing.

This type of study can yield an upper-bound estimate of the highest level of modeling accuracy that might be attainable given the available information—as long as one can assume that the human "wizards" are more competent at the type of assessment in question than a fully automatic system is likely to be in the foreseeable future. In this example study, the expert advisers showed some ability to identify the users' goals and needs; if they had not done so, perhaps the entire project would have been reconsidered.

Aside from accuracy, a Wizard-of-Oz study can also shed light on problems with the acceptability and usability of the new system, as long as they concern the content and basic nature of the adaptations performed, as opposed to interface details that are not faithfully reproduced in the study. In this example study, it turned out that even incorrect advice was often taken seriously by the users, who wasted time following up on irrelevant suggestions. One design implication is that it may be worthwhile to make users aware of the fact that the system's advice is not necessarily relevant.

#### Comparisons With the Work of Human Designers

Just as humans can sometimes be employed as a surrogate for a user-adaptive system in the early stages of design, humans can sometimes also serve as a standard of comparison for the evaluation of an implemented system. This method makes most sense when the system is performing a task at which human authors or designers are likely to be experienced and skilled, such as the tailoring of the content of a

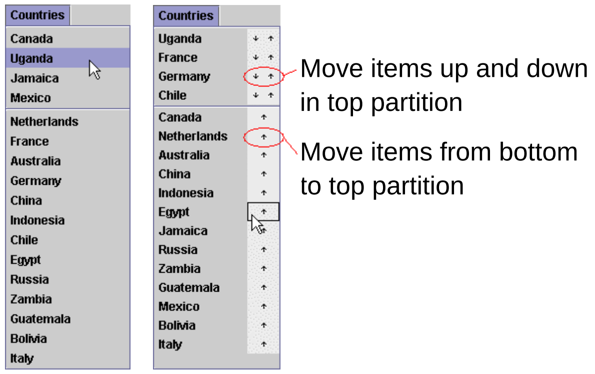


**Figure 22.27.** The adaptive (left) and adaptable (right) split menus used in an experimental comparison. (Adapted from Figure 1 of "A comparison of static, adaptive, and adaptable menus," by L. Findlater & J. McGrenere, 2004, in E. Dykstra-Erickson & M. Tscheligi (Eds.), *Human factors in computing systems: CHI 2004 conference proceedings*, pp. 89–96, New York: ACM, Copyright 2004 by the Association for Computing Machinery, Inc. Adapted with permission.)

presentation to the individual user. An example of such a study was mentioned earlier: a comparison of the information displays generated by the real estate recommender system RIA with those generated by two experienced designers. Instead of performing the same task as the system, the human designers may simply act as judges of the appropriateness of the system's output. In either case, the comments of the designers can yield valuable qualitative information to complement the objective results.

#### Experimental Comparisons of Adaptive and Nonadaptive Systems

Many experimental studies involving user-adaptive systems compare an adaptive variant with some nonadaptive one. This strategy is understandable given the doubt that often exists as to whether the additional overhead required for user-adaptivity is justified by any sort of improvement in the interaction. But studies like this are trickier to conduct and interpret than they may seem at first glance.

As an example, consider the experiment by Findlater and McGrenere (2004), which examined whether subjects could work faster using (a) an adaptive menu system somewhat like SMART MENUS (cf. the discussion of adaptive menus above); (b) an adaptable system in which the users explicitly determined the content of the menus themselves; or (c) a conventional static menu system. For each of these three types of menu, a realization was chosen that seemed optimal in the context of the experiment. All three menus were a special type of *split menu* (Sears & Shneiderman, 1994): The four most frequent items in each menu were placed in a special section at the top of the menu for quick access (instead of being temporarily hidden, as in SMART MENUS). The items selected for the upper part of the static menu were optimally chosen in that they reflected the actual frequency of the items in the experimental tasks. The adaptive menu was initially identical to the static menu, but the arrangement of the items changed as a function of the user's behavior, favoring the most frequently and recently used menu items. For the adaptable menu, the upper part was initially empty, so that users would be encouraged to perform some adaptations. Though the overall pattern of results is complex, it tends to speak in favor of the adaptable menu. But note that the conditions did not give the adaptive variant much of a chance to provide any benefit: Since the initial menu was already the best possible single menu for the experimental tasks, adaptation could improve performance only by taking advantage of any local concentrations of commands within a particular period of time (e.g., the need to execute the same command several times in succession). By contrast, in normal usage situations, an adaptive menu can also improve performance by reflecting increasingly the user's longer-term patterns of use.

The difficulties in interpreting the results of this experiment could not easily have been avoided with a different design: Any other way of realizing the three conditions would have left some different set of questions open. The lesson of this and many other examples is that comparisons between adaptive and nonadaptive variants of a system should not be viewed as empirical tests whose results can be interpreted straightforwardly. Instead, they should be seen as shedding light on various aspects of the ways in which people use adaptive and nonadaptive systems and on the effectiveness of these methods in certain conditions.

#### Taking Into Account Individual Differences

Individual differences among users show up in just about every user study in the field of human-computer interaction; but with the user-adaptive systems they are especially important, because of the wide range of subjective reactions that user-adaptivity tends to invoke (illustrated, for example, by the different preferences reported by Findlater & McGrenere, 2004, for different participants in the experiment discussed in the previous subsection). As a result, asking whether people like a particular type of user-adaptive system is in many cases like asking whether the voters in a given country prefer progressive or conservative policies. Even if, in a given sample, a statistically significant tendency in one direction or the other can be found, important minority points of view should be understood and reported. As in politics, the goal should be to take into account the range of different preferences in a way that is satisfactory to at least a large proportion of the potential user group.

When individual differences are present, it may be tempting to try to find correlations with demographic characteristics or with general personality variables. Some relationships of this sort can be found (see, for example, Graziola, Pianesi, Zancanaro, & Goren-Bar, 2005, with regard to personality variables), but it is not always worthwhile to focus much attention on them. The relationships tend to be weak, since differing responses can also be due to more specific causes such as the degree of familiarity with the type of system in question and the particular conditions under which a user performs a task.

### Checking Usability Under Realistic Conditions

With just about any type of interactive system, new lessons are likely to be learned when a working prototype (or the finished system itself) is tested in realistic situations, even if the system has been studied thoroughly in earlier stages. With user-adaptive systems, realistic testing is especially advisable because of the issues discussed in the section on usability challenges, whose importance for a given system can often be assessed only in real use. For example, an obtrusive proactive recommender might be considered quite acceptable, or even amusing, when the user is performing some artificial assigned task in a laboratory; in the real world, when she is under pressure to complete an important task quickly, the interruptions may be evaluated quite differently. Similarly, privacy issues are serious mainly when real data about the user are involved, whose misuse could have real consequences.

## THE FUTURE OF USER-ADAPTIVE SYSTEMS

This chapter has shown that adaptive interfaces, agents, and other user-adaptive systems do not represent a smooth and easy shortcut to more successful human-computer interaction: They present a complex set of usability challenges, and they require carefully designed methods of acquiring information about users, as well as relatively sophisticated computational techniques that are not needed in other types of interactive system. Even when all of these requirements have been dealt with, it is often tricky to prove empirically that user-adaptivity has actually added any value. It is no wonder that some experts believe that the interests of computer users are better served by continued progress within more familiar paradigms of user-centered system design.

On the other hand, our understanding of the complex challenges raised by user-adaptive systems has been growing steadily, and they are now familiar and valued elements in a number of types of system, as the survey in the first two major sections of this chapter has shown.

### Growing Need for User-Adaptivity

Increases in the following variables suggest that the functions served by user-adaptivity will continue to grow in importance:

*Diversity of Users and Contexts of Use* Computing devices are being used by an ever-increasing variety of users in an increasing variety of contexts. It is therefore becoming harder to design a system that will be suitable for all users and contexts without some sort of user-adaptivity or user-controlled adaptability; and as has been discussed at several points in this chapter, user-controlled adaptability has its limitations.

*Number and Complexity of Interactive Systems* The functions of user-adaptivity discussed in the first major section of this chapter involve helping users to deal effectively with interactive systems and tasks even when they are not able or willing to gain complete understanding and control in each individual case. This goal becomes increasingly important as the number—and in some cases the complexity—of the systems that people must deal with continues to increase—because of factors ranging from the growth of the world-wide web to the proliferation of miniature interactive computing devices.

*Scope of Information to Be Dealt With* Even when using a single, relatively simple system, users today can often access a much larger and more diverse set of objects of interest than they could a few years ago—be they documents, products, or potential collaborators. It is therefore becoming relatively more attractive to delegate some of the work of dealing with these objects—even to a system which has an imperfect model of the user's requirements. In the early 1990s, the idea that an email sorting agent such as the one described by Maes (1994) might delete an incoming message without consulting the user seemed preposterous to many people. After the huge increase in the amount of (largely unwanted) email that has occurred since then, many people now regularly allow dozens of their incoming messages to be deleted unseen.

### Increasing Feasibility of Successful Adaptation

As the need for user-adaptivity increases, so—fortunately—does its feasibility, largely because of advances in the following areas:

*Ways of Acquiring Information About Users* Most of the methods discussed in the section about acquiring information about users are becoming more powerful with advances in technology and research. They therefore offer the prospect of substantial increases in the quality of adaptation—although methods for ensuring users' privacy call for equal attention.

*Advances in Techniques for Learning, Inference, and Decision* In addition to the more general progress in the fields of machine learning and artificial intelligence, communities of researchers have been focusing on the specific requirements of computational techniques that support user-adaptivity. Consequently, noticeable progress is being made every year.

*Attention to Empirical Methods* The special empirical issues and methods that are involved in the design and evaluation of user-adaptive systems have been receiving increasing attention from researchers, as emphasis has shifted from high technical sophistication to ensuring that the systems enhance the users' experience.

Despite these tendencies, it is actually unlikely that the number of deployed systems associated with labels like "user-adaptive" will increase. Once an adaptation technique has left the research laboratory and started playing some genuinely useful role in people's lives, it tends to be described in terms of the function that it serves rather than in terms of the techniques that it uses. Awareness of the commonalities discussed in this chapter should help both to increase the number of systems that succeed in this way and to recognize them despite the new labels that are placed on them.

## ACKNOWLEDGEMENTS

Preparation of this chapter was supported by the German Ministry of Education and Research (projects SPECTER and COLLATE). The following colleagues provided valuable assistance in the form of comments on earlier drafts, supplementary information about their research, and/or graphics depicting their systems: Vikram Aggarwal and Michelle Zhou; James 'Bo' Begole and Nicholas Matsakis; Vania Dimitrova; Tatsuya Kawahara; Bruce Krulwich; Greg Linden; Eric McCreath and Judy Kay; Joanna McGrenere and Leah Findlater; Antonija Mitrovic; Charles Rich and Elyon DeKoven; Yasuyuki Sumi and Kenji Mase; Karin Vogel.

## REFERENCES

*Note:* Since the relevant literature through 2001 was covered by the earlier version of this chapter in the first edition of the *Human-Computer Interaction Handbook* (Jameson, 2003), the references given here are mostly from the period between the first and second edition. Readers with a strong interest in the topic of this chapter are advised to consult the version of this chapter from the first edition as well, as many of the works cited there still represent important contributions to the field.

Aberg, J., & Shahmehri, N. (2001). An empirical study of human web assistants: Implications for user support in web information systems. In J. A. Jacko, A. Sears, M. Beaudouin-Lafon, & R. J. Jacob (Eds.), *Human factors in computing systems: CHI 2001 conference proceedings* (pp. 400–411). New York: ACM.

Bälter, O., & Sidner, C. L. (2002). Bifrost Inbox Organizer: Giving users control over the inbox. In *Proceedings of NordiCHI 2002* (pp. 111–118).

Bartlett, M. S., Littlewort, G., Fasel, I., & Movellan, J. R. (2003). Real time face detection and facial expression recognition: Development and applications to human computer interaction. In *Proceedings of the Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction at the 2003 Conference on Computer Vision and Pattern Recognition* (pp. 53–58).

Beaudouin-Lafon, M., & Mackay, W. E. (2008). Prototyping tools and techniques. In A. Sears & J. A. Jacko (Eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (pp. 1017–1039). Boca Raton, FL: CRC Press.

Billsus, D., Hilbert, D., & Maynes-Aminzade, D. (2005). Improving proactive information systems. In J. Riedl, A. Jameson, D. Billsus, & T. Lau (Eds.), *IUI 2005: International Conference on Intelligent User Interfaces* (pp. 159–166). New York: ACM.

Billsus, D., & Pazzani, M. J. (2007). Adaptive news access. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 550–572). Berlin: Springer.

Bontcheva, K., & Wilks, Y. (2005). Tailoring automatically generated hypertext. *User Modeling and User-Adapted Interaction*, *15*, 135–168.

Brusilovsky, P., Kobsa, A., & Nejdl, W. (Eds.). (2007). *The adaptive web: Methods and strategies of web personalization.* Berlin: Springer.

Budzik, J., Hammond, K., & Birnbaum, L. (2001). Information access in context. *Knowledge-Based Systems*, *14*, 37–53.

Bunt, A., Carenini, G., & Conati, C. (2007). Adaptive content presentation for the web. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 409–432). Berlin: Springer.

Bunt, A., Conati, C., & McGrenere, J. (2004). What role can adaptive support play in an adaptable system? In C. Rich & N. J. Nunes (Eds.), *IUI 2004: International Conference on Intelligent User Interfaces* (pp. 117–124). New York: ACM.

Burke, R. D., Hammond, K. J., & Young, B. C. (1997). The FindMe approach to assisted browsing. *IEEE Expert*, *12*(4), 32–40.

Cawsey, A., Grasso, F., & Paris, C. (2007). Adaptive information for consumers of healthcare. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 465–484). Berlin: Springer.

Cheverst, K., Byun, H. E., Fitton, D., Sas, C., Kray, C., & Villar, N. (2005). Exploring issues of user model transparency and proactive behaviour in an office environment control system. *User Modeling and User-Adapted Interaction*, *15*(3–4), 235–273.

Corbett, A. (2001). Cognitive computer tutors: Solving the two-sigma problem. In M. Bauer, P. Gmytrasiewicz, & J. Vassileva (Eds.), *UM2001, User Modeling: Proceedings of the Eighth International Conference* (pp. 137–147). Berlin: Springer.

Corbett, A. T., Koedinger, K. R., & Anderson, J. R. (1997). Intelligent tutoring systems. In M. Helander, T. K. Landauer, & P. V. Prabhu (Eds.), *Handbook of human-computer interaction (2nd ed.)* (pp. 849–874). Amsterdam: North-Holland.

Cranor, L. F. (2004). 'I didn't buy it for myself': Privacy and ecommerce personalization. In C.-M. Karat, J. Blom, & J. Karat (Eds.), *Designing personalized user experiences in eCommerce* (pp. 57–74). Dordrecht, Netherlands: Kluwer.

Crawford, E., Kay, J., & McCreath, E. (2002). An intelligent interface for sorting electronic mail. In Y. Gil & D. B. Leake (Eds.), *IUI 2002: International Conference on Intelligent User Interfaces* (pp. 182–183). New York: ACM.

DeKoven, E. A. (2004). *Help Me Help You: Designing support for person-product collaboration.* Unpublished doctoral dissertation, Delft University of Technology.

Fernandez, R., & Picard, R. W. (2000). Modeling drivers' speech under stress. In *Proceedings of the ISCA Workshop on Speech and Emotions.* Belfast.

Findlater, L., & McGrenere, J. (2004). A comparison of static, adaptive, and adaptable menus. In E. Dykstra-Erickson & M. Tscheligi (Eds.), *Human factors in computing systems: CHI 2004 conference proceedings* (pp. 89–96). New York:

ACM.

Fogarty, J., Ko, A. J., Aung, H. H., Golden, E., Tang, K. P., & Hudson, S. E. (2005). Examining task engagement in sensor-based statistical models of human interruptibility. In W. Kellogg, S. Zhai, G. van der Veer, & C. Gale (Eds.), *Human factors in computing systems: CHI 2005 conference proceedings* (pp. 331–340). New York: ACM.

Fuller, R. (2005). Towards a general theory of driver behaviour. *Accident Analysis & Prevention*, *37*(3), 461–472.

Gena, C., & Weibelzahl, S. (2007). Usability engineering for the adaptive web. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 720–766). Berlin: Springer.

Gervasio, M. T., Moffitt, M. D., Pollack, M. E., Taylor, J. M., & Uribe, T. E. (2005). Active preference learning for personalized calendar scheduling assistance. In J. Riedl, A. Jameson, D. Billsus, & T. Lau (Eds.), *IUI 2005: International Conference on Intelligent User Interfaces* (pp. 90–97). New York: ACM.

Graziola, I., Pianesi, F., Zancanaro, M., & Goren-Bar, D. (2005). Dimensions of adaptivity in mobile systems: Personality and people's attitudes. In J. Riedl, A. Jameson, D. Billsus, & T. Lau (Eds.), *IUI 2005: International Conference on Intelligent User Interfaces* (pp. 223–230). New York: ACM.

Gruen, D., Rohall, S. L., Minassian, S., Kerr, B., Moody, P., Stachel, B., Wattenberg, M., & Wilcox, E. (2004). Lessons from the ReMail prototypes. In J. Herbsleb & G. Olson (Eds.), *Proceedings of the 2004 Conference on Computer-Supported Cooperative Work* (pp. 152–161). New York: ACM.

Gutkauf, B., Thies, S., & Domik, G. (1997). A user-adaptive chart editing system based on user modeling and critiquing. In A. Jameson, C. Paris, & C. Tasso (Eds.), *User modeling: Proceedings of the Sixth International Conference, UM97* (pp. 159–170). Vienna: Springer Wien New York.

Haller, S., McRoy, S., & Kobsa, A. (Eds.). (1999). *Computational models of mixed-initiative interaction.* Dordrecht, The Netherlands: Kluwer.

Hammond, N. (1987). Principles from the psychology of skill acquisition. In M. M. Gardiner & B. Christie (Eds.), *Applying cognitive psychology to user-interface design* (pp. 163–188). Chichester, England: Wiley.

Hanson, V. L., & Crayne, S. (2005). Personalization of web browsing: Adaptations to meet the needs of older adults. *Universal Access in the Information Society*, *4*, 46–58.

Healey, J., & Picard, R. (2000). SmartCar: Detecting driver stress. In *Proceedings of the Fifteenth International Conference on Pattern Recognition* (pp. 4218–4221). Barcelona.

Hegner, S. J., McKevitt, P., Norvig, P., & Wilensky, R. L. (Eds.). (2001). *Intelligent help systems for UNIX*. Dordrecht, the Netherlands: Kluwer.

Ho, J., & Intille, S. S. (2005). Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In W. Kellogg, S. Zhai, G. van der Veer, & C. Gale (Eds.), *Human factors in computing systems: CHI 2005 conference proceedings* (pp. 909–918). New York: ACM.

Höök, K. (2000). Steps to take before IUIs become real. *Interacting with Computers*, *12*(4), 409–426.

Horvitz, E. (1999). Principles of mixed-initiative user interfaces. In M. G. Williams, M. W. Altom, K. Ehrlich, & W. Newman (Eds.), *Human factors in computing systems: CHI 1999 conference proceedings* (pp. 159–166). New York: ACM.

Horvitz, E., Breese, J., Heckerman, D., Hovel, D., & Rommelse, K. (1998). The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In G. F. Cooper & S. Moral (Eds.), *Uncertainty in artificial intelligence: Proceedings of the fourteenth conference* (pp. 256–265). San Francisco: Morgan Kaufmann.

Horvitz, E., Koch, P., Sarin, R., Apacible, J., & Subramani, M. (2005). Bayesphone: Precomputation of context-sensitive policies for inquiry and action in mobile devices. In L. Ardissono, P. Brna, & A. Mitrovic (Eds.), *UM2005, User modeling: Proceedings of the Tenth International Conference* (pp. 251–260). Berlin: Springer.

Iqbal, S. T., Adamczyk, P. D., Zheng, X. S., & Bailey, B. P. (2005). Towards an index of opportunity: Understanding changes in mental workload during task execution. In W. Kellogg, S. Zhai, G. van der Veer, & C. Gale (Eds.), *Human factors in computing systems: CHI 2005 conference proceedings* (pp. 311–318). New York: ACM.

Jacko, J. A., Leonard, V. K., & Scott, I. (2008). Perceptual impairments: New advancements promoting technological access. In A. Sears & J. A. Jacko (Eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (pp. 853–870). Boca Raton, FL: CRC Press.

Jameson, A. (2003). Adaptive interfaces and agents. In J. A. Jacko & A. Sears (Eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (pp. 305–330). Mahwah, NJ: Erlbaum.

Jameson, A., & Schwarzkopf, E. (2002). Pros and cons of controllability: An empirical study. In P. De Bra, P. Brusilovsky, & R. Conejo (Eds.), *Adaptive hypermedia and adaptive web-based systems: Proceedings of AH 2002* (pp. 193–202). Berlin: Springer.

Jameson, A., & Smyth, B. (2007). Recommendation to groups. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 596–627). Berlin: Springer.

Jones, C. M., & Jonsson, I.-M. (2005). Automatic recognition of affective cues in the speech of car drivers to allow appropriate responses. In *Proceedings of OZCHI 2005.*

Karat, J., Karat, C.-M., & Brodie, C. (2008). Human-computer interaction viewed from the intersection of privacy, security, and trust. In A. Sears & J. A. Jacko (Eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (pp. 639–658). Boca Raton, FL: CRC Press.

Kelly, D., & Teevan, J. (2003). Implicit feedback for inferring user preference: A bibliography. *ACM SIGIR Forum*, *37*(2), 18–28.

Kobsa, A. (2007). Generic user modeling systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 136–154). Berlin: Springer.

Kobsa, A., & Wahlster, W. (Eds.). (1989). *User models in dialog systems.* Berlin: Springer.

Komatani, K., Ueno, S., Kawahara, T., & Okuno, H. G. (2005). User modeling in spoken dialogue systems to generate flexible guidance. *User Modeling and User-Adapted Interaction, 15*(1–2), 169–183.

Krüger, A., Baus, J., Heckmann, D., Kruppa, M., & Wasinger, R. (2007). Adaptive mobile guides. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 521–549). Berlin: Springer.

Krulwich, B. (1997). Lifestyle Finder: Intelligent user profiling using large-scale demographic data. *AI Magazine*, *18*(2), 37–45.

Lai, J., Karat, C.-M., & Yankelovich, N. (2008). Conversational speech interfaces and technologies. In A. Sears & J. A. Jacko (Eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (pp. 381–391). Boca Raton, FL: CRC Press.

Langley, P., & Fehling, M. (1998). *The experimental study of adaptive user interfaces* (Tech. Rep. No. 98-3). Institute for the Study of Learning and Expertise, Palo Alto, CA.

Lanier, J. (1995). Agents of alienation. *interactions*, *2*(3), 66–72.

Lesh, N., Rich, C., & Sidner, C. L. (1999). Using plan recognition in human-computer collaboration. In J. Kay (Ed.), *UM99, User modeling: Proceedings of the Seventh International Conference* (pp. 23–32). Vienna: Springer Wien New York.

Li, X., & Ji, Q. (2005). Active affective state detection and user assistance with dynamic Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, *35*(1), 93–105.

Linden, G., Hanks, S., & Lesh, N. (1997). Interactive assessment of user preference models: The automated travel assistant. In A. Jameson, C. Paris, & C. Tasso (Eds.), *User modeling: Proceedings of the Sixth International Conference, UM97* (pp. 67–78). Vienna: Springer Wien New York.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, *7*(1), 76–80.

Liscombe, J., Riccardi, G., & Hakkani-Tür, D. (2005). Using context to improve emotion detection in spoken dialog systems. In *Proceedings of the Ninth European Conference on Speech Communication and Technology* (pp. 1845–1848). Lisbon.

Lisetti, C. L., & Nasoz, F. (2004). Using noninvasive wearable computers to recognize human emotions from physiological signals. *EURASIP Journal on Applied Signal Processing*, *11*, 1672–1687.

Litman, D. J., & Pan, S. (2002). Designing and evaluating an adaptive spoken dialogue system. *User Modeling and User-Adapted Interaction*, *12*(2–3), 111–137.

Looi, C.-K., McCalla, G., Bredeweg, B., & Breuker, J. (Eds.). (2005). *Artificial intelligence in education: Supporting learning through intelligent and socially informed technology.* Amsterdam: IOI Press.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM*, *37*(7), 30–40.

McCarthy, K., Reilly, J., McGinty, L., & Smyth, B. (2005). Experiments in dynamic critiquing. In J. Riedl, A. Jameson, D. Billsus, & T. Lau (Eds.), *IUI 2005: International Conference on Intelligent User Interfaces* (pp. 175–182). New York: ACM.

McCreath, E., & Kay, J. (2003). i-ems: Helping users manage email. In P. Brusilovsky, A. Corbett, & F. de Rosis (Eds.), *UM2003, User modeling: Proceedings of the Ninth International Conference* (pp. 263–272). Berlin: Springer.

McCreath, E., Kay, J., & Crawford, E. (2005). i-ems - an approach that combines hand-crafted rules with learnt instance-based rules. *Australian Journal of Intelligent Information Processing Systems*, *9*(1), 49–63.

McGrenere, J., Baecker, R. M., & Booth, K. S. (2002). An evaluation of a multiple interface design solution for bloated software. In L. Terveen, D. Wixon, E. Comstock, & A. Sasse (Eds.), *Human factors in computing systems: CHI 2002 conference proceedings* (pp. 164–170). New York: ACM.

Micarelli, A., Gasparetti, F., Sciarrone, F., & Gauch, S. (2007). Personalized search on the world wide web. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 195–230). Berlin: Springer.

Mitchell, T., Caruana, R., Freitag, D., McDermott, J., & Zabowski, D. (1994). Experience with a learning personal assistant. *Communications of the ACM, 37*(7), 81–91.

Mitrovic, A., & Ohlsson, S. (1999). Evaluation of a constraint-based tutor for a database language. *International Journal of Artificial Intelligence in Education*, *10*, 238–256.

Mitrovic, A., Suraweera, P., Martin, B., & Weerasinghe, A. (2004). DB-Suite: Experiences with three intelligent, web-based database tutors. *Journal of Interactive Learning Research*, *15*(4), 409–432.

Mobasher, B. (2007). Data mining for web personalization. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 90–135). Berlin: Springer.

Müller, C., Wittig, F., & Baus, J. (2003). Exploiting speech for recognizing elderly users to respond to their special needs. In *Proceedings of the Eighth European Conference on Speech Communication and Technology* (pp. 1305–1308). Geneva.

Norman, D. A. (1994). How might people interact with agents? *Communications of the ACM*, *37*(7), 68–71.

Ozok, A. A. (2008). Survey design and implementation in HCI. In A. Sears & J. A. Jacko (Eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (pp. 1151–1169). Boca Raton, FL: CRC Press.

Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, *13*, 393–408.

Petrelli, D., & Not, E. (2005). User-centred design of flexible hypermedia for a mobile guide: Reflections on the HyperAudio experience. *User Modeling and User-Adapted Interaction, 15*(3/4), 303–338.

Picard, R. W. (1997). *Affective computing.* Cambridge, MA: MIT Press.

Rhodes, B. J. (2000). *Just-in-time information retrieval.* Unpublished doctoral dissertation, School of Architecture and Planning, Massachusetts Institute of Technology, Cambridge, MA. (Available from http://rhodes.www.media.mit.edu/people/rhodes/research/)

Rich, C., Sidner, C., Lesh, N., Garland, A., Booth, S., & Chimani, M. (2005). DiamondHelp: A collaborative interface framework for networked home appliances. In *In 5th International Workshop on Smart Appliances and Wearable Computing, IEEE International Conference on Distributed Computing Systems Workshops* (pp. 514–519).

Rich, C., & Sidner, C. L. (1998). COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, *8*, 315–350.

Rich, C., Sidner, C. L., & Lesh, N. (2001). COLLAGEN: Applying collaborative discourse theory to human-computer interaction. *AI Magazine*, *22*(4), 15–25.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 291–324). Berlin: Springer.

Schiele, B., Starner, T., Rhodes, B., Clarkson, B., & Pentland, A. (2001). Situation aware computing with wearable computers. In W. Barfield & T. Caudell (Eds.), *Fundamentals of wearable computers and augmented reality* (pp. 511–538). Mahwah, NJ: Erlbaum.

Schneider, M., & Kiesler, S. (2005). Calling while driving: Effects of providing remote traffic context. In W. Kellogg, S. Zhai, G. van der Veer, & C. Gale (Eds.), *Human factors in computing systems: CHI 2005 conference proceedings* (pp. 561–570). New York: ACM.

Sears, A., & Shneiderman, B. (1994). Split menus: Effectively using selection frequency to organize menus. *ACM Transactions on Computer-Human Interaction*, *1*, 27–51.

Sears, A., Young, M., & Feng, J. (2008). Physical disabilities and computing technologies: An analysis of impairments. In A. Sears & J. A. Jacko (Eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (pp. 821–852). Boca Raton, FL: CRC Press.

Segal, R. B., & Kephart, J. O. (1999). MailCat: An intelligent assistant for organizing e-mail. In *Proceedings of the Third International Conference on Autonomous Agents* (pp. 276–282).

Segal, R. B., & Kephart, J. O. (2000). Incremental learning in SwiftFile. In P. Langley (Ed.), *Machine Learning: Proceedings of the 2000 International Conference* (pp. 863–870). San Francisco: Morgan Kaufmann.

Shearin, S., & Lieberman, H. (2001). Intelligent profiling by example. In J. Lester (Ed.), *IUI 2001: International Conference on Intelligent User Interfaces* (pp. 145–151). New York: ACM.

Shneiderman, B., & Maes, P. (1997). Direct manipulation vs. interface agents. *interactions*, *4*(6), 42–61.

Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M., & Boydell, O. (2005). Exploiting query repetition and regularity in an adaptive community-based web search engine. *User Modeling and User-Adapted Interaction*, *14*(5), 383–423.

Soller, A. (2007). Adaptive support for distributed collaboration. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 573–595). Berlin: Springer.

Sumi, Y., & Mase, K. (2001). AgentSalon: Facilitating face-to-face knowledge exchange through conversations among personal agents. In *Proceedings of the Fifth International Conference on Autonomous Agents* (pp. 393–400). Montreal.

Sumi, Y., & Mase, K. (2002). Supporting the awareness of shared interests and experiences in communities. *International Journal of Human-Computer Studies*, *56*(1), 127–146.

Surendran, A. C., Platt, J. C., & Renshaw, E. (2005). Automatic discovery of personal topics to organize email. In *Proceedings of the Second Conference on Email and Anti-Spam.* Stanford, CA.

Teevan, J., Dumais, S., & Horvitz, E. (2005). Personalizing search via automated analysis of interests and activities. In *Proceedings of 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 449–456). Salvador, Brazil.

Terveen, L., & McDonald, D. W. (2005). Social matching: A framework and research agenda. *ACM Transactions on Computer-Human Interaction*, *12*(3), 401–434.

Trewin, S. (2004). Automating accessibility: The Dynamic Keyboard. In *Proceedings of ASSETS 2004* (pp. 71–78). Alanta, GA.

Tsandilas, T., & schraefel, m. (2004). Usable adaptive hypermedia. *New Review of Hypermedia and Multimedia*, *10*(1), 5–29.

VanLehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., & Wintersgill, M. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education*, *15*(3), 147–204.

Vivacqua, A., & Lieberman, H. (2000). Agents to assist in finding help. In T. Turner, G. Szwillus, M. Czerwinski, & F. Paternò (Eds.), *Human factors in computing systems: CHI 2000 conference proceedings* (pp. 65–72). New York: ACM.

Wainer, H. (Ed.). (2000). *Computerized adaptive testing: A primer* (2nd ed.). Hillsdale, NJ: Erlbaum.

Wexelblat, A., & Maes, P. (1997). *Issues for software agent UI.* (Unpublished manuscript, available from http://citeseer.ist.psu.edu/128549.html on September 20th, 2009. Cited with permission.)

Wickens, C. D., & Hollands, J. G. (2000). *Engineering psychology and human performance* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Wilde, G. J. (1998). Risk homeostasis theory: An overview. *Injury Prevention*, *4*(2), 89–91.

Wolfman, S. A., Lau, T., Domingos, P., & Weld, D. S. (2001). Mixed initiative interfaces for learning tasks: SMARTedit talks back. In J. Lester (Ed.), *IUI 2001: International Conference on Intelligent User Interfaces* (pp. 167–174). New York: ACM.

Yacoub, S., Simske, S., Lin, X., & Burns, J. (2003). Recognition of emotions in interactive voice response systems. In *Proceedings of the Eighth European Conference on Speech*

*Communication and Technology* (pp. 729–732). Geneva.

Zhou, M. X., & Aggarwal, V. (2004). An optimization-based approach to dynamic data content selection in intelligent multimedia interfaces. In *Proceedings of UIST 2004* (pp. 227–236).

Zhou, M. X., Wen, Z., & Aggarwal, V. (2005). A graph-matching approach to dynamic media allocation in intelligent multimedia interfaces. In J. Riedl, A. Jameson, D. Billsus, & T. Lau (Eds.), *IUI 2005: International Conference on Intelligent User Interfaces* (pp. 114–121). New York: ACM.

Ziegler, C.-N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. In *Proceedings of the 2003 International World Wide Web Conference* (pp. 22–32).

Zukerman, I., & Litman, D. (2001). Natural language processing and user modeling: Synergies and limitations. *User Modeling and User-Adapted Interaction*, *11*, 129–158.