Authors' prefinal version (June 2011); not to be posted on web.

Systems That Adapt to Their Users

Anthony Jameson and **Krzysztof Z. Gajos**
DFKI, German Research Center for Artificial Intelligence
Harvard School of Engineering and Applied Sciences

Introduction

This chapter covers a broad range of interactive systems which have one idea in common: that it can be worthwhile for a system to learn something about individual users and adapt its behavior to them in some nontrivial way.

A representative example is shown in Figure 20.1: the CommunityCommands recommender plug-in for AutoCAD (introduced by Matejka, Li, Grossman, & Fitzmaurice, 2009, and discussed more extensively by Li, Matejka, Grossman, Konstan, & Fitzmaurice, 2011). To help users deal with the hundreds of commands that AutoCAD offers—of which most users know only a few dozen—CommunityCommands (a) gives the user easy access to several recently used commands, which the user may want to invoke again soon; and (b) more proactively suggests commands that this user has not yet used but may find useful, given the type of work they have been doing recently.

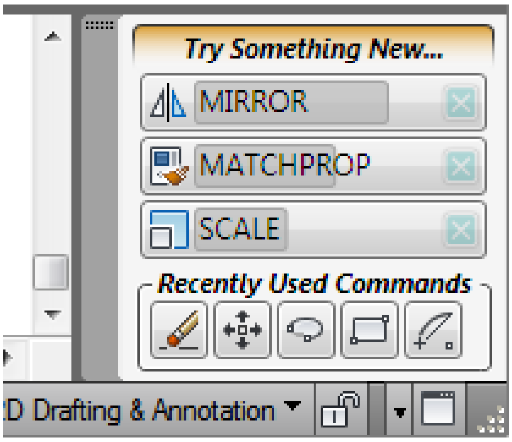


Figure 20.1: Screenshot showing how CommunityCommands recommends commands to a user.

(Image supplied by Justin Matejka. The length of the darker bar for a command reflects its estimated relevance to the user's activities. When the user hovers over a command in this interface, a tooltip appears that explains the command and might show usage hints provided by colleagues. See also http://www.autodesk.com/research.)

Concepts

A key idea embodied in CommunityCommands and the other systems discussed in this chapter is that of *adaptation to the individual user*. Depending on their function and form, particular types of systems that adapt to their users have been given labels including *adaptive user interfaces*, *software agents*, *recommender systems*, and *personalization*. 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 20.2 introduces some concepts that can be applied to any user-adaptive system; Figure 20.3 shows the form that they take in recommendations generated by CommunityCommands.

A user-adaptive system makes use of some type of information about the current individual user, such as the commands that the user has executed. 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 their pattern of command use). 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; this process may be straightforward, or it can involve some fairly sophisticated decision making on the part of the system.

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.

The second half of the definition is necessary because otherwise any interactive system could be said to "adapt" to its users, even if it just responds straightforwardly to key presses. It is the processes of user model acquisition and application that raise many common issues and challenges that characterize user-adaptive systems.

This definition also distinguishes user-adaptive systems from purely *adaptable* systems: ones that offer the user an opportunity to configure or otherwise influence the system's longer-term behavior (for example, by choosing options that determine the appearance of the user interface). Often, what works best is a carefully chosen combination of adaptation and adaptability. For example, if the user of CommunityCommands is not interested in the command MATCHPROP, she can click on the "close" button next to it to specify

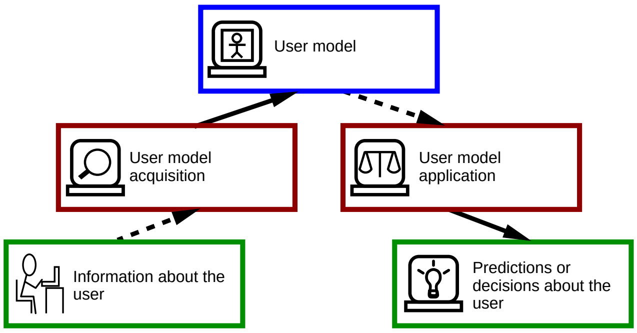


Figure 20.2: General schema for the processing in a user-adaptive system.
(Dotted arrows: use of information; solid arrows: production of results.)

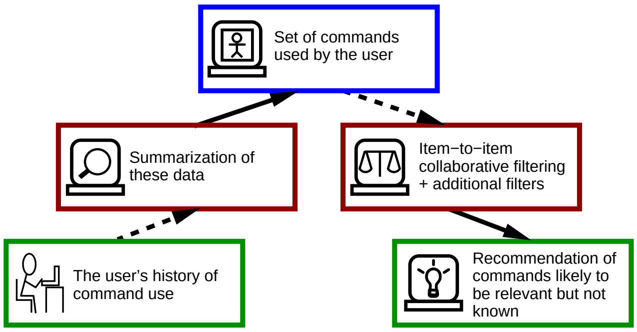


Figure 20.3: Overview of adaptation in COMMUNITYCOMMANDS.

that it should not be recommended again. Keeping the user "in the loop" in this way can be an essential part of effective and well-accepted adaptation.

Chapter Preview

The next two sections of this chapter address the question "What can user-adaptivity be good for?" They examine in turn a number of different functions that can be served by user-adaptivity, giving examples ranging from familiar commercially deployed systems to research prototypes. The subsequent section discusses some usability challenges that are especially important in connection with user-adaptive systems, challenges which stimulated much of the controversy that surrounded these systems when they first began to appear in the 1980s and 1990s. The next section considers a key design decision: What types of information about each user should be collected? The chapter concludes with a reflection on the current state of the art and the future challenges for user-adaptive systems..[1]

[1]Interested readers may also want to consult the chapters on this topic in the first two editions of this handbook (Jameson, 2003, 2008), which include discussions of earlier user-adaptive systems that can still serve as instructive examples, as well as discussions of typical issues and methods associated with empirical studies of user-adaptive systems.

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 four types of support.

Adaptively Offering Help

The first form is the one illustrated by the COMMUNITYCOMMANDS recommender: In cases where it is not sufficiently obvious to users how they should operate a given application, a help system can adaptively offer information and advice about how to use it—and perhaps also execute some actions on behalf of the user. That is, the system can act like a helpful friend who is looking over the user's shoulder–a service which users often greatly appreciate but which is not in general easy to automate effectively. The adaptation can make the help which is offered more relevant to the user's needs than the more commonly encountered user-independent help.

The main way in which COMMUNITYCOMMANDS helps the user is by recommending possibly useful commands that the user has not yet employed. The basic recommendation technique is *collaborative filtering*, which is discussed later in this chapter in connection with systems that recommend products. The central idea is: "People who use commands like the ones that you have been using also use the following commands, which you may not be familiar with: . . . ."

Matejka et al. (2009) explain how the basic collaborative filtering algorithm had to be adapted and supplemented to yield good performance for command recommendation. For example, if a user already knows the command *A*, it makes little sense to recommend a command *B* which is just a similar or less efficient way of achieving the same effect; so handcrafted rules were added to prevent such recommendations.

Experience with the deployment of COMMUNITYCOMMANDS as an AUTOCAD plug-in has indicated that this approach appears to have general feasibility and usefulness for systems that offer a large number of commands. This case study can also be seen as a successful application of the strategy of looking for a relatively lightweight approach to adaptation that still offers considerable added value. Attention to the details of the adaptive algorithms and of the user interface design appears to be more important here than the use of more complex adaptation technology.

Systems that offer help in an adaptive way have a long history. Perhaps the most obvious—but also the most difficult—scenario is one in which a user is trying to achieve a particular goal (e.g., align the objects in a drawing in a particular way) but does not know how to achieve the goal with the system. A helper could in principle automatically recognize the user's difficulty and suggest a way of solving the problem. 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.[2] During the 1990s, such work became less frequent, perhaps partly because of a recognition of the fundamental difficulties involved: It is in general hard to recognize what goal a user is pursuing when the user is not performing actions that serve their goal. And spontaneously offering help can be distracting, since the system cannot be sure that the user is interested in getting help. 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).

For these reasons, more recent research has focused on less ambitious but still potentially useful ways of adaptively offering help. A strategy in this category—one which is is quite different from that of COMMUNITYCOMMANDS—is to view the process of offering help as involving collaboration and dialog with the user. A representative of this paradigm is the DIAMONDHELP system, which assists users in the operation of complex consumer devices (see, e.g., Rich et al., 2005). DIAMONDHELP is somewhat reminiscent of the (nonadaptive) "wizards" that walk users through procedures such as the configuration of new software; but is more flexible and adaptive in that it applies a *mixed-initiative* paradigm, allowing the user to perform sequences of actions on her own if she likes and trying to keep track of what she is doing. Rich (2009) offers a recent discussion of this general paradigm.

Taking Over Parts of Routine Tasks

Another 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. Two traditionally popular candidates for automation of this sort (discussed briefly below) have been the sorting of email and the scheduling of appointments and meetings.

The system TASKTRACER illustrates a number of typical functionalities of systems in this category.[3] The tedious work that is taken over by TASKTRACER is not a single, separate chore but rather parts of many of the routine subtasks that are involved in everyday work with a normal desktop (or laptop) computer. The central insight is that a user is typically multitasking among a set of *projects*, each of which is associated with a diverse set of *resources*, such as files, web pages, and email messages. Since these resources tend to be stored in different places and used by different applications,

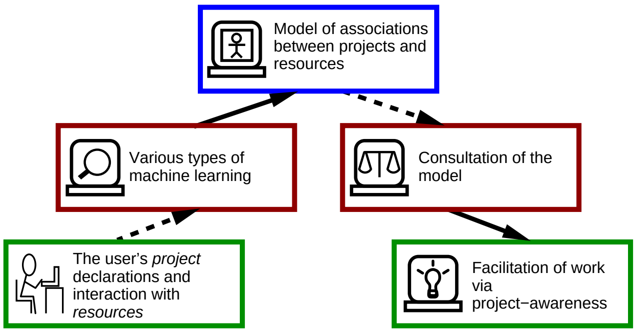


Figure 20.5: Overview of adaptation in TASKTRACER.

a significant proportion of everyday computer work involves locating and accessing the resources that are relevant to the project that is currently in the focus of attention.

The user of TASKTRACER creates a structured list of projects that they sometimes work on; once they have done so, the system does two things largely autonomously: (a) By observing the user, it learns which resources are associated with which projects. (b) It tries to figure out which project the user is working on at any given moment (see, e.g., Shen, Irvine, et al., 2009) As can be seen in Figure 20.5, these two functions constitute the adaptive aspects of the system.

Even if these inferences by the system are not entirely accurate, they can help the user in various ways: For example, when the user wants to save a document that they have created, TASKTRACER can save them some mouse clicks by suggesting 2 or 3 folders associated with the current project in which they might like to store the new file. And when a user switches to a project, TASKTRACER can offer a list of the resources associated with the current project, sorted by recency of access, so that the user can quickly locate them again (see, e.g., Figure 20.4).

A more difficult form of support that still represents a challenge involves supporting the user in executing *workflows* (see, e.g., Shen, Fitzhenry, & Dietterich, 2009). That is, instead of just recognizing that the user is working on the project "quarterly report", the system (a) learns by observation what steps are involved in the preparation of a quarterly report; (b) keeps track of how much of the quarterly report workflow the user has executed so far; and (c) supports the user in remembering and executing subsequent steps. The tendency of users to multitask makes this type of support potentially valuable, but it also makes it challenging for systems to do the necessary learning and activity tracking.

Two traditionally popular candidates for automation of this sort have been sorting or filtering email and scheduling appointments and meetings. Classic early research on these tasks included the work of Pattie Maes's group on "agents that reduce work and information overload" (see, e.g., Maes, 1994). Another perennially studied task in this category is the scheduling of meetings and appointments (T. Mitchell,

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

[3] See Dietterich, Bao, Keiser, and Shen (2010), for a recent comprehensive discussion of TASKTRACER and http://eecs.oregonstate.edu/TaskTracer/ for further information and references.

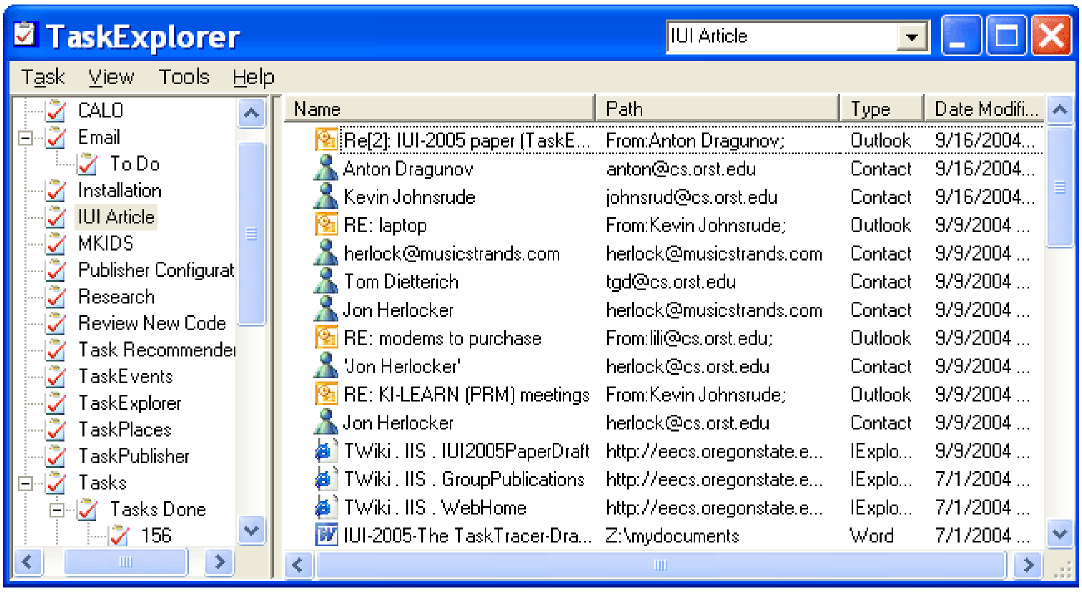


Figure 20.4: Screenshot showing one of the ways in which TASKTRACER helps a user to find resources associated with a given *project*.

(Here, the various resources associated with "IUI Article" are listed in order of recency. Image retrieved from http://eecs.oregonstate.edu/TaskTracer/ in March 2011, reproduced with permission of Thomas Dietterich.)

Caruana, Freitag, McDermott, & Zabowski, 1994; Horvitz, 1999; Gervasio, Moffitt, Pollack, Taylor, & Uribe, 2005): By learning the user's general 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.

Systems of this sort can actually take over two types of work from the user: 1. choosing what particular action is to be performed (e.g., which folder a file should be saved in); and 2. performing the mechanical steps necessary to execute that action (e.g., clicking in the file selector box until the relevant folder has been reached). Adaptation to the user is required only for the first type of work; but the second type of work cannot be performed without the first type.

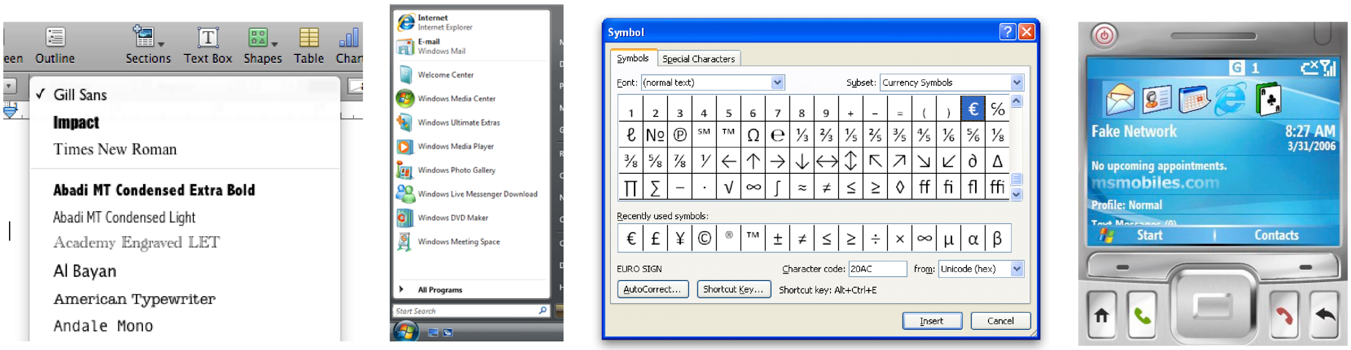
In the ideal case, the system could make the correct choice with such confidence that it would not even be necessary to consult the user, and the entire task would be automated, with the user perhaps not even being aware that it was being performed. In many cases, though, the user does have to be involved in the choice process, because the system can only help to make the choice, not make it autonomously. In these cases, the amount of mental and physical work saved is much lower. Hence there is a trade-off between the amount of control that the user has over the choices being made and the amount of effort they save. Users can differ as to where they want to be on this trade-off curve at any given time, depending on factors like the importance of making a correct choice and the amount of other work that is competing for their attention. The typical pattern is for users to begin by exercising careful control over the performance of the task and then to relinquish control gradually to the system, as the system's competence increases (because of learning) and/or the user becomes better able to predict what the system will be able to do successfully. Trade-offs of this sort will be discussed in the section on usability challenges.

Adapting the Interface to Individual Tasks and Usage

A different way of helping a person to use a system more effectively is to adapt the presentation and organization of the interface so that it fits better with the user's tasks and usage patterns. The potential benefit of this type of adaptation is that it can improve the user's motor performance by bringing functionality that is likely to be used closer or making interface elements larger; improve perceptual performance by making relevant items easier to find; or improve cognitive performance by reducing complexity.

An example of this type of an adaptive interface that will be familiar to most readers is the font selection menus available in popular productivity software. Figure 20.6(a) illustrates the basic mechanism: The most recently selected items are copied to the top part of the menu. This top part, clearly visually separated from the rest of the menu, holds the adaptive content. If a desired font is present in the top section, the user can select it either from that section or from its usual location in the lower part of the menu.

The concept generalizes beyond menus; it can be used to adapt many different types of user interface component, as is illustrated in Figure 20.6. We use the term *split interfaces* to



(a) (b) (c) (d)

Figure 20.6: Examples of modern implementations of adaptive Split Interfaces.

(a: The most recently used fonts are copied to the clearly designated adaptive top part of the menu in APPLE PAGES*. A user wishing to select the Times New Roman font, has the option of either taking advantage of the adaptation or following the familiar route to the usual location of that font in the main part of the menu. b: Recently or frequently used programs are copied to the main part of the* WINDOWS *7 start menu while also remaining accessible through the "All Programs" button. c: Recently used special symbols are copied to a separate part of the dialog box in the symbol chooser in* MS OFFICE *2007. d: Recently used applications are easily accessible on a* WINDOWS MOBILE *phone.)*

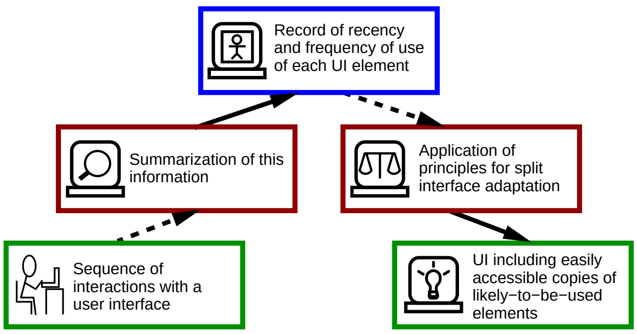


Figure 20.7: Overview of adaptation in Split Interfaces.

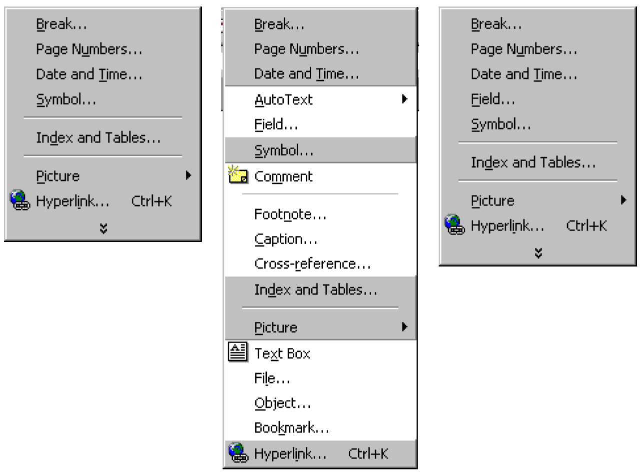


Figure 20.8: In Microsoft SMART MENUS, rarely used items are removed from the menu thus reducing the apparent complexity of the application.

(Hovering over the menu or clicking on the double arrow below the last item causes the menu to be expanded showing the elided items. If a hidden item is selected by the user, it is immediately visible on the subsequent visits to that menu.)

refer to the successful general design pattern in which adaptation is used to copy functionality predicted to be most relevant to the user to a designated adaptive part of the user interface.

Several studies have demonstrated that split interfaces reliably improve both satisfaction and performance (Findlater & McGrenere, 2008; K. Z. Gajos, Czerwinski, Tan, & Weld, 2006). What makes split interfaces successful is that they offer an effort saving to those users who are willing to take advantage of the adaptation while not upsetting the familiar routine for those who prefer to use the basic interface consistently.

Designs that require users to alter their behavior are often rejected. A widely known example of an early adaptive interface that elicited mixed reactions from users is the SMART MENUS that Microsoft introduced in WINDOWS 2000 (Figure 20.8; see McGrenere, Baecker, & Booth, 2007, for an extensive comparison of this type of adaptation with user-controlled customization). To reduce the apparent complexity of the software, these menus were designed to show only a subset of the features—the most basic ones and those that the user used frequently or recently. The remaining features were shown if the user dwelled on a menu without selecting anything or if he clicked on a downward pointing arrow at the bottom of the menu. The design had the promise of simplifying the interaction most of the time for most users, but for some users the confusion caused when trying to find infrequently used functionality outweighed the potential benefits.

An early illustrative example involves automatically reordering menu items on the basis of the frequency of use (J. Mitchell & Shneiderman, 1989). This approach resulted in poorer performance and lower user satisfaction than the nonadaptive baseline. In this case, the lack of success of the adaptive strategy can be attributed to the fact that because of the constantly changing order of menu items, users could never reach the level of visual search efficiency predicted by Hick-Hyman law (Hick, 1952) for familiar interfaces.

A radically different approach to menu adaptation—called *ephemeral adaptation*—was introduced recently by Findlater, Moffatt, McGrenere, and Dawson (2009). In ephemeral adaptation, the menu items that are predicted to be most likely to be selected by the user are displayed immediately when the menu is opened, while the remaining items fade in gradually over a short period of time (e.g., 500 ms). This adaptation takes advantage of the fact that an abrupt appearance of a new object involuntarily captures our attention, while its gradual onset does not. Because the user's attention is drawn to the small subset of items that are shown immediately when the menu opens, it is easy for users to locate these items quickly. This adaptive mechanism focuses entirely on users' visual search performance. It has been demonstrated to improve overall performance without increasing selection times for the items that are gradually faded in.

A user-driven alternative to the class of adaptive approaches described in this section is customization. Customization, however, requires significant upfront effort on user's part and consequently very few people choose to customize their interfaces (Mackay, 1991; Palen, 1999) and even fewer re-customize them as their needs change (McGrenere et al., 2007). Mixed initiative approaches (e.g., Bunt, Conati, & McGrenere, 2007) that combine the two approaches show promise for providing good balance between efficiency and user control.

The adaptive designs discussed in this section were prototyped and evaluated mostly with menus and toolbars, but the underlying concepts can be generalized to a broader range of settings. Findlater and Gajos (2009) provide a more in-depth exploration of the design space of user interfaces that adapt to users' tasks.

Adapting the Interface to Individual Abilities

Next we consider systems that adapt their user interfaces to the *abilities* of their users.

The promise of this type of adaptation is that it can provide personalized experience to people whose needs with respect to the user interface are unique, variable over time, or hard to anticipate. This is precisely the situation of the many users with impairments. Not only are these users different from the "average" user, they are also significantly different from each other: even people with very similar diagnoses can have very different actual abilities (Bergman & Johnson, 1995; Hwang, Keates, Langdon, & Clarkson, 2004; Keates, Langdon, Clarkson, & Robinson, 2002; Law, Sears, & Price, 2005). Currently, these users have to adapt themselves—often using specialized assistive technologies—to the existing user interfaces. Adaptive systems offer the possibility to reverse this situation: why not adapt user interfaces to the unique needs and abilities of people with impairments?

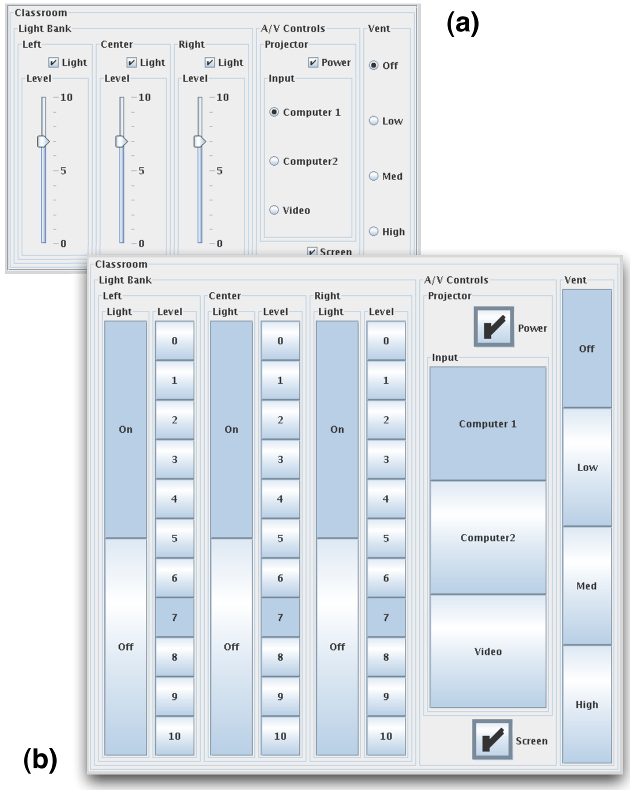


Figure 20.9: Example of the ability-based adaptation in SUPPLE: (a) the default interface for controlling lighting and A/V equipment in a classroom; (b) a user interface for the same application automatically generated by SUPPLE for a user with impaired dexterity based on a model of her actual motor abilities.

Impairments do not have to be permanent or to be a result of a medical condition. For example, environmental factors such as temperature may temporarily impair a person's dexterity; a low level of illumination will impact reading speed; and ambient noise will affect hearing ability. These factors are particularly relevant to mobile computing. Indeed, studies have shown that in relation to standing still, walking results in lower pointing speed and accuracy, as well as decreased reading speed and comprehension (Barnard, Yi, Jacko, & Sears, 2007; Lin, Goldman, Price, Sears, & Jacko, 2007). These results suggest that there is both a need and an opportunity to adapt mobile interaction to the momentary effective abilities of users.

The SUPPLE system (Gajos, Wobbrock, ^{a} Weld, 2007, 2008, 2010) provides an example of ability-based adaptation for people with motor impairments. SUPPLE requires each user to perform a one-time set of diagnostic tasks so that the system can build a model of that person's unique motor abili-

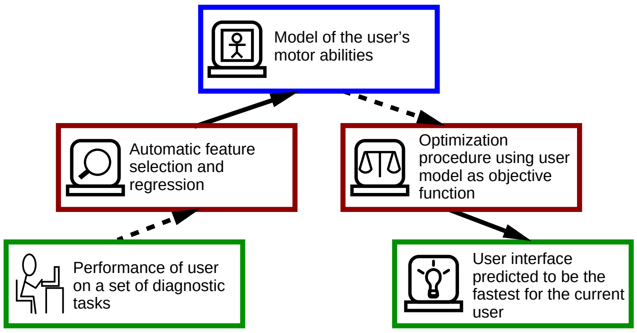


Figure 20.10: Overview of adaptation in SUPPLE.

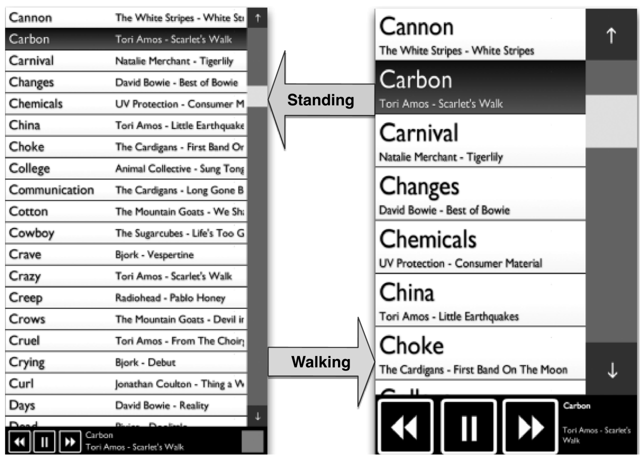


Figure 20.11: The Walking UI—an example of an adaptation to a temporary situationally-induced impairment. The larger buttons address the decreased pointing speed and accuracy of walking users; the larger fonts for song titles help with impaired reading speed; and the differences in font sizes between titles and additional song information help direct fragmented attention.

(Screen shots courtesy of Shaun Kane.)

ties. After that, for any application the user wants to interact with, SUPPLE uses optimization methods to automatically generate user interfaces that are predicted to be the fastest to use for this person. Figure 20.9 shows an example of a dialog box automatically generated by SUPPLE for a user with impaired dexterity due to a spinal cord injury. The results of an experiment involving 11 participants with a variety of motor impairments demonstrate that the automatically generated interfaces that were adapted to users' individual motor abilities resulted in significantly improved speed, accuracy, and satisfaction (see, e.g., K. Gajos, Wobbrock, & Weld, 2008). On the average, these interfaces helped close over 60% of the performance gap between able-bodied users and users with motor impairments.

The WALKING UI prototype (Kane, Wobbrock, & Smith, 2008) shown in Figure 20.11 provides an example of what an adaptation to the changing abilities of mobile users might look like. The UI has two versions, one for when the user is stationary and one for when they are in motion. The two versions follow a very similar design to ensure that the users do not have to learn two separate user interfaces. The walking variant has larger interactors to compensate for users' impaired dexterity, larger fonts for song titles to accommodate reduced reading ability, and a more visually salient presentation for song titles than for secondary information to mitigate the effects of fragmented attention.

These types of system have been evaluated in laboratory studies, but since they have not yet been widely deployed, we cannot yet provide empirical evidence showing what the main challenges to adoption of these systems are. But several such challenges can be anticipated: Obtaining useful models of users' abilities while placing minimum burden on the users is clearly one such challenge. The studies evaluating the SUPPLE system demonstrated that models created from direct measurements of users' abilities resulted in significantly more successful interfaces than those that were based on users' expressed preferences, but those direct measurements of abilities required users to go through a one-time but hour-long set of diagnostic tasks. Another factor that seems likely to have an impact on adoption of interfaces like that of Figure 20.11 is the method for controlling the switch between different user interface variants. A fully manual approach is likely to be found too inefficient, while one that is fully automated may cause confusion.

Wobbrock, Kane, Gajos, Harada, and Froelich (2011) present several other examples of ability-based interfaces, discuss the rationale for ability-based design, and propose a set of guiding principles.

Functions: Supporting Information Acquisition

Even back in the days when computers were chained to desktops, people were complaining about information overload and clamoring for tools that would help them to focus their attention on the documents, products, and people that really mattered to them. Since then, the flood has grown to a tsunami. Two of the most conspicuous developments have been (a) mobile devices that enable people to produce and consume information wherever they are; and (often in combination with these) social networks, which are increasingly replacing face-to-face communication.

This information overload constitutes a powerful motivation for the development of systems that adapt to their users: Computers have the technical capability to reduce the information tsunami to a trickle that people can manage; but since people are generally not interested in the same trickle, computers can do so effectively only by taking into account properties of the user such as their interests, current tasks, and context.

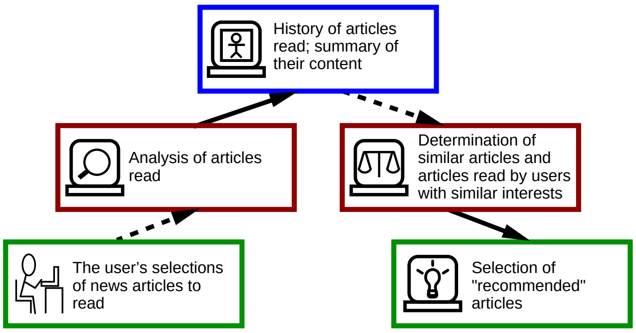


Figure 20.13: Overview of adaptation in GOOGLE's personalized news recommendations.

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.

One type of document that has become more pervasive over the past several years comprises news reports of the type traditionally found in printed newspapers. With so many news sources now available online, the amount of choices available to a person who wants to read a few interesting news reports has increased drastically, even when we take into consideration the fact that one report may turn up in a number of variations in different news sources.

One news website that addresses this problem is GOOGLE NEWS. One of the solutions offered by the site is a section of "recommended" stories and that are selected on the basis of the users' previous clicks on other news stories within the same site (see Figure 20.12). The first personalization algorithms used for this purpose by GOOGLE were based on collaborative filtering, which is found in several other systems discussed in this chapter. But as is described by Liu, Dolan, and Pedersen (2010), it proved necessary to include some *content-based* filtering as well, recommending stories related to general themes that the current user had previously shown an interest in. In particular, it is otherwise hard to recommend hot-off-the-press news stories that have not yet attracted many clicks from other users. Roughly speaking, in this application the content-based filtering helps by revealing what topics the current user is generally interested in, while the collaborative filtering helps to keep track of temporary trends (e.g., a surge of interest in the newly released iPad) that apply to larger groups of users and that are likely to be followed to some extent by any given individual user as well.

More generally speaking, user-adaptive systems that help users find information [4] typically draw from the vast repertoire of techniques for analyzing textual information (and to a lesser but increasing 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 GOOGLE NEWS:

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) describe 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 assessments of the user's interests on the part of the system.

Support for Query-Based Search

When a user is just checking the latest news or casually browsing for interesting information, the user is not in general expressing a specific information need. Hence it is relatively easy for a user model to help noticeably by presenting information that is especially likely to be of interest to this particular user. By contrast, when a user formulates an explicit query, as in a web search engine, it is less obvious how a user model can help to identify relevant information. And in fact, the *potential for personalization* (Teevan, Dumais, & Horvitz, 2010) has been found to vary considerably from one query to the next. If just about all of the users who issue a given query end up choosing the same documents from those returned by the system, there is little that an individual user model can do to increase the usefulness of the search results. But for queries that tend to result in very different selections for different users (e.g., the query "chi"), *personalized search* can add value. The basic idea is that the list of search results that would normally be returned is reordered (or *biased*) on the basis of a user model, which is in turn based on some aspects of the user's previous behavior with the system. GOOGLE has offered personalized search on its main search engine for several years—though many users are probably unaware of the personalization, which tends not to change the ranking of the search results in an immediately noticeable way for most queries.

The idea of assessing the potential for personalization is

[4] 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).

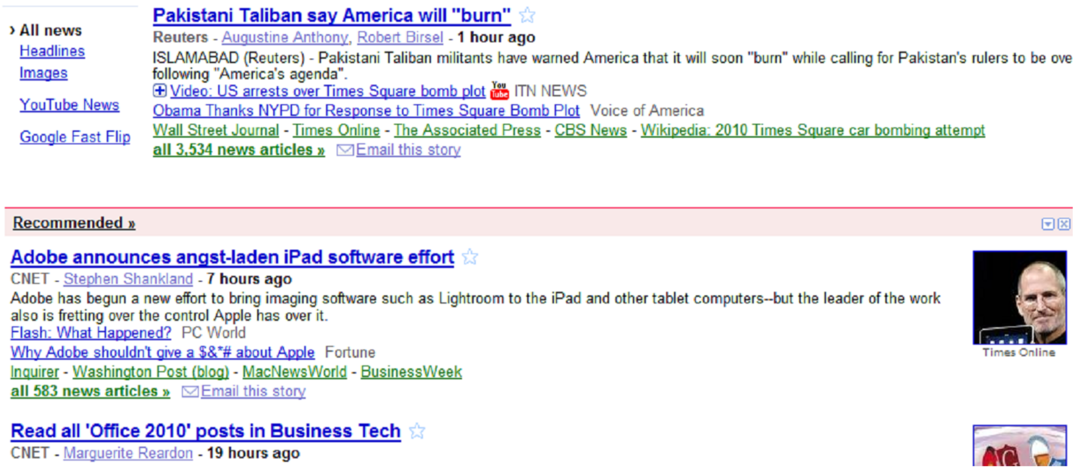


Figure 20.12: A small section of a front page of GOOGLE NEWS, including the personalized section.
(This user had recently selected a number of computer-related articles, and at this time the newly launched iPad was a popular topic.)

worth considering with other forms of adaptation as well: If we can estimate in advance the possible benefits of adaptation, perhaps before designing or implementing any adaptive mechanism, we can more efficiently identify situations in which the benefits of adaptation will outweigh the costs.

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 retrieve information relevant to it from external sources. An illustrative recent example[5] is the AMBIENT HELP system (Matejka, Grossman, & Fitzmaurice, 2011), which can also be seen as an approach to the problem of offering adaptive help that was discussed at the beginning of this chapter: While a user works with a complex application on one computer monitor, AMBIENT HELP uses a second monitor to display videos and texts with tutorial material that has some relevance to the user's current working context. 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 information that is offered and derive little or no benefit from the system. AMBIENT HELP expands the space of design solutions by introducing an unobtrusive way of showing what a video has to offer (with a dimmed image, a reduced frame rate, and muted volume) and a scheme for allowing users quickly to explore the content of the available videos. Previous work in the same vein (e.g., by Billsus, Hilbert, & Maynes-Aminzade, 2005) suggests allowing users to adjust the relative obtrusiveness of the proactively offered information to suit their individual taste.

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

Figure 20.14: Part of a screen showing a movie recommendation generated on request by netflix.com.
(Screen shot made from http://netflix.com/ in March 2010.)

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 primary benefit of these systems is that they assist users in navigating large collections of products by surfacing items that are both novel and relevant.

An example that will be familiar to most readers is shown in Figure 20.14. A visitor to NETFLIX.com has just explicitly requested recommendations, without having specified a par-

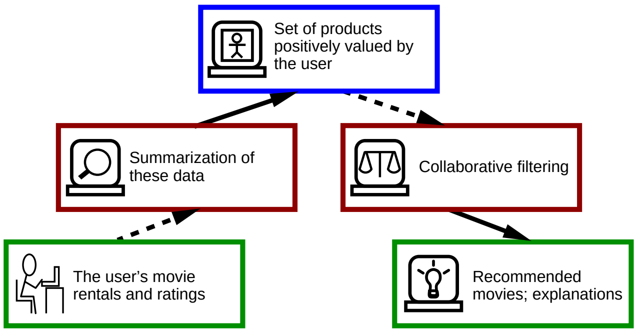


Figure 20.15: Overview of adaptation in NETFLIX.

ticular type of a movie. During the user's past visits, NETFLIX has learned about his interests, on the basis of movies he has watched and ratings he has made. Therefore, the system can make recommendations that are especially likely to appeal to this particular user.

The recommendations of NETFLIX embody many design decisions that contribute to the success of this type of adaptation:

First, as can be inferred from the brief explanations that accompany the recommendation in Figure 20.14, the system takes as a starting point the information it has about the user's prior viewing history and ratings. It then compares these with the ratings of other users to generate predictions for the current user. That is, the recommendations are based on a statistical analysis of ratings made by many users, an approach known as *collaborative filtering* (see, for example, Schafer, Frankowski, Herlocker, & Sen, 2007, for a general overview).[6] The products recommended in this way may also happen to be similar in the sense of belonging to the same genre or having the same director, but similarities of this sort can also be conspicuously absent: In the example in Figure 20.14, a nature documentary is recommended to a customer based on his past enjoyment of Kurosawa's lighthearted samurai story "Yojimbo." The power of collaborative filtering comes from the observation that many features relevant to our choices are hard to capture. In the movie domain, for example, the mood, the particular style of humor, or the details of the camera work, may be as relevant as the more easily describable properties such as genre, director, or the cast.

Second, the explanations accompanying the recommendation are another important design feature: For example, Taking into account the fact that what is "good" often depends on context (for example, a user may enjoy a complex drama one day, while preferring a less demanding action movie after a long work day), the explanations help users better predict if a particular film is what they are looking for at a given moment. As in the example in Figure 20.14, many sites use other items the user has rated in the past as a basis for generating an explanation. But very different types of information can also be used for explanations, such as user-generated tags (Vig, Sen, & Riedl, 2009). A recent discussion of the many forms that explanations can take and the functions that they can serve has been provided by Tintarev and Masthoff (2010).

Finally, movie recommendations in NETFLIX complement rather than replace the normal searching and browsing capabilities. This property allows users to decide which mode of interaction is most appropriate in their situation.

Many products, such as movies, vacations, or restaurant meals, are often enjoyed by groups rather than individuals, 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; the explanations of the recommendations may have to refer to the preferences of the individual group members; and it may be worthwhile for the system to help the users negotiate to arrive at a final decision on the basis of the recommendations.

Another design challenge for recommender systems has to do with the availability of information about the users' preferences. Collaborative filtering is less effective for supporting infrequent decisions such as a digital camera purchase, which can involve one-time considerations that are not closely related to previous choices by the same user. Since the 1980s, researchers have worked on systems that explicitly elicit information about the user's needs (and the tradeoffs among them) and help the user identify products that best meet their needs. One particularly effective interaction paradigm for such systems is *example critiquing* (see, e.g., Burke, Hammond, & Young, 1997, for an early exposition and Pu & Chen, 2008, for a discussion 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.

Finally, a highly pervasive and economically vital form of product recommendation is advertising. Over the last decade, on-line advertising has shifted largely from attention-grabbing banners and pop-ups to subtler *personalized* ads. Rather than relying on users' explicit feedback in the form of purchases and product ratings (as is the case with recommender systems), on-line personalized advertising relies on implicit input such as the search terms, contents of an email message (in the case of GMAIL ads), the topics of the pages visited, and the browsing history. There are many

[6] Interested readers will also find many documents available on the web about the highly publicized efforts of NETFLIX to encourage improvement of its algorithms by sponsoring the NETFLIX Prize.

good reasons to prefer such personalized advertising: it tends to be presented in a less intrusive way (e.g., the text-only ads used by GOOGLE) and it has the promise of being more relevant to the users. Indeed, a recent study found that of people who clicked on personalized ads, twice as many were likely to actually make a purchase than people who clicked on non-personalized ads (Beales, 2010).

However, because on-line behavioral data (such as searches the people perform and sites that they visit) are considered sensitive personal information, and because the users do not have clear and effective means of controlling what information they divulge to advertisers and when, privacy concerns about personalized advertising are common (Federal Trade Commission, 2009). The Canadian Marketing Association (2009) has found that only about 30% of North Americans are comfortable with advertisers tracking their browsing behavior for the purpose of providing more targeted advertising, even though nearly half like seeing ads for coupons and promotions from online stores and brands that they have purchased from before. Improving the comprehensibility of and user control over data collection are therefore important challenges for the long-term success of personalized advertising (cf. the discussion of usability challenges later in this chapter).

Tailoring Information Presentation

The previous two sections discussed systems that help users decide *what* information (such as news items or product descriptions) to consider. We now turn to systems that adapt *how* information is presented.

A striking and practically important example is found in the work of Jefferson and Harvey (2006, 2007), which uses personalized models of color perception abilities of color-blind users to adapt the presentation of graphical information in a way that preserves the saliency and readability of color-encoded information. A major challenge in adapting content to the individual color perception abilities is that complex color-encoded information needs to be conveyed through a reduced color palette. One possible approach is to generate a fixed mapping that tries to "squeeze" the full spectrum of visible colors into a range that is distinguishable by a particular individual. This approach inevitably reduces perceptual differences among the colors in the transformed palette. Instead,Jefferson and Harvey (2006) compute these mappings for each image individually. Their approach takes advantage of the fact that most images use only a limited number of colors for salient information. Their algorithm automatically identifies these salient colors and computes a mapping from the original palette to one that is appropriate for the user. The mapping is computed in a way that preserves the perceptual differences among the important colors.

Unfortunately, because the process of computing an optimal color mapping is computationally expensive—up to several minutes may be required—it is not feasible for interactive

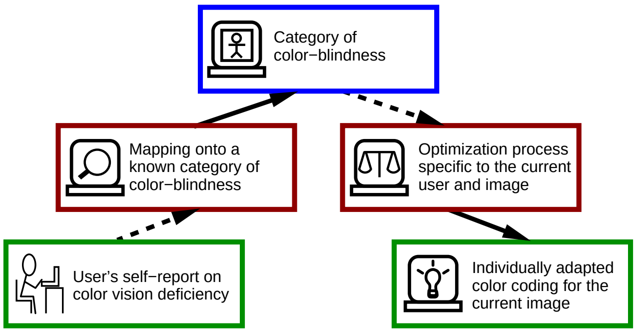


Figure 20.16: Overview of Jefferson and Harvey's method of adaptation to color-blindness.

use. Jefferson and Harvey (2007) have developed an alternative approach where the computer quickly generates a small set of possible mappings that may be appropriate for a particular individual and the user can quickly select the appropriate one with a slider, while getting an immediate preview of the effect. By splitting the adaptation burden between the computer and the user, this particular system provides users with a solution that is effective and fast, and requires only a minimal amount of manual effort to use.

The remaining challenge is that of quickly creating accurate models of individual color perception abilities. Fortunately, most users can be helped adequately by being stereotyped into one of a small number of discrete color blindness categories. However, some types of color blindness (the anomalous trichromacies) form a spectrum from almost normal color perception to almost complete inability to distinguish certain pairs of colors. While there exist some methods for building models of individual color perception abilities (e.g., Brettel, Viénot, & Mollon, 1997, Gutkauf, Thies, & Domik, 1997), they require that users engage in an explicit diagnostic task, and one that may need to be repeated for different display devices. A faster, unobtrusive method is still needed.

Tailoring often concerns information in textual form. An important application area here comprises systems that present medical information to patients, who may differ greatly in terms of their interest in and their ability to understand particular types of information (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).

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 for a broad overview).

Bringing People Together

One of the most striking changes in computer use over the past several years has been the growth of social networks. Whereas people used to complain about being overwhelmed by the number of emails and other documents that they were expected to read, they can now also be overwhelmed by the number of comments posted on their social network homepage, the number of people who would like to link up with them—and even the suggestions that they get from sites like FACEBOOK and LINKEDIN concerning possible social links. Accordingly, personalized support for decisions about whom to link up with has become a practically significant application area for user-adaptive systems.

Figure 20.17 shows how an internal social networking site used at IBM called SOCIALBLUE (formerly BEEHIVE) recommends a colleague who might be added to the user's network.

As the example illustrates, SOCIALBLUE makes extensive use of information about social relationships to arrive at recommendations: not just information about who is already explicitly linked with whom in the system (which is used, for example, on FACEBOOK) but also types of implicit information that are commonly available within organizations, such as organizational charts and patent databases.

As described by J. Chen, Geyer, Dugan, Muller, and Guy (2009), SOCIALBLUE also uses information about the similarity between two employees (e.g., the overlap in the words used in available textual descriptions of them).

These authors found that these two types of information tend to lead to different recommendations, which in turn are accepted or rejected to differing extents and for different reasons. For example, information about social relationships works better for finding colleagues that the current user already knows (but has not yet established a link to in the system), while information about similarity is better for finding promising unknown contacts.

Taking the analysis of the same data a step further, Daly, Geyer, and Millen (2010) showed that different algorithms can also have different consequences for the structure of the social network in which they are being used. For example, a system that recommends only "friends of friends" will tend to make the currently well-connected members even better connected. This result illustrates why it is often worthwhile to consider not only how well an adaptive algorithm supports a user in a typical individual case but also what its broader, longer-term consequences may be.

Given that the various contact recommendation algorithms can be used in combination in various ways, a natural conclusion is that designers of systems of this sort should consider what mix of the algorithm types makes most sense for their particular system and application scenario.

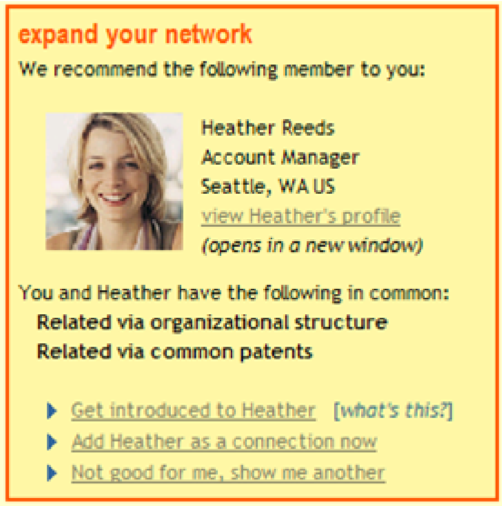


Figure 20.17: Screenshot of from SOCIALBLUE showing how it recommends a potentially interesting colleague.
(Image retrieved from http://www-users.cs.umn.edu/ jilin/projects.html in March 2011, reproduced with permission of Jilin Chen and Werner Geyer.)

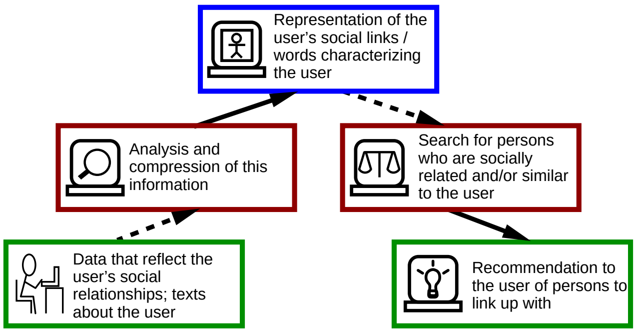


Figure 20.18: Overview of adaptation in SOCIALBLUE.

Other contexts in which some sort of social matching has proved useful include:

- Expert finding, which involves identifying a person who has the knowledge, time, and social and spatial proximity that is necessary for helping the user to solve a particular problem (see, e.g., Shami, Yuan, Cosley, Xia, & Gay, 2007; Ehrlich, Lin, & Griffiths-Fisher, 2007; Terveen & McDonald, 2005).
- Recommendation of user communities that a user might like to join—or at least use as an information resource (see, e.g., W.-Y. Chen, Zhang, & Chang, 2008, Carmagnola, Vernero, & Grillo, 2009, and Vasuki, Natarajan, Lu, & Dhillon, 2010) for early contributions to this relatively novel problem.
- Collaborative learning, which has become a popular approach in computer-supported learning environments (see,

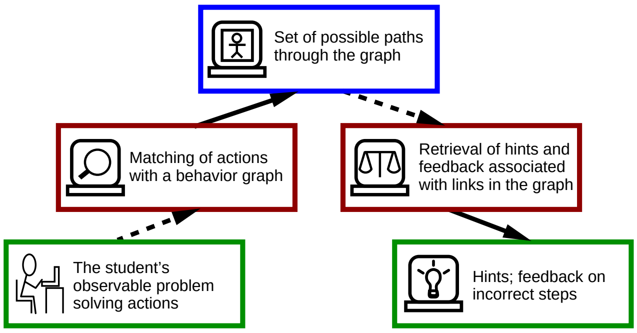


Figure 20.20: Overview of adaptation in the STOICHIOMETRY TUTOR.

e.g., Soller, 2007).

Supporting Learning

Some of the most sophisticated forms of adaptation to users have been found in tutoring systems and learning environments: systems designed to supplement human teaching by enabling students to learn and practice without such teaching while still enjoying some of its benefits.[7]

An illustrative recent example is the web-based STOICHIOMETRY TUTOR (Figure 20.19; McLaren, DeLeeuw, & Mayer, 2011a, McLaren et al.), which helps students to practice solving elementary chemistry problems using basic mathematics. In the example shown, the student must perform a unit conversion and take into account the molecular weight of alcohol. The interface helps to structure the student's thinking, but it is still possible to make a mistake, as the student in the example has done by selecting "H20" instead of "COH4" in the lower part of the middle column. Part of the system's adaptation consists in hints that it gives when the student makes a mistake (or clicks on the "Hint" link in the upper right). The key knowledge that underlies the adaptation is a *behavior graph* for each problem: a representation of acceptable paths to a solution of the problem, along with possible incorrect steps. Essentially, the tutor is like a navigation system that knows one or more ways of getting from a specified starting point to a destination; but instead of showing the student a "route" to follow, it lets the user try to find one, offering hints when the student makes a wrong turn or asks for advice. This approach enables the system to adapt with some flexibility: It can deal with multiple strategies for solving the problem and entertain multiple interpretations about the student's behavior.

This relatively recent approach to tutoring is called *example tracing* (Aleven, McLaren, Sewall, & Koedinger, 2009), because it involves tracing the student's progress through the behavior graph, which in turn represents, in generalized form, a set of examples of how a problem can be solved. For authors of tutoring systems, providing such examples is a relatively easy, practical way to give the system the knowledge that it needs to interpret the student's behavior. In the long history of systems that adaptively support learning, most systems have employed more complex representations of the to-be-acquired knowledge and of the student's knowledge current state of knowledge (for example, in terms of sets of rules or constraints). Example tracing is an instance of a general trend to look for simpler but effective ways of achieving useful adaptation, relative to the often complex ground-breaking systems that are developed in research laboratories.

Giving feedback and hints about steps in solving a problem is an example of *within-problem guidance*, sometimes called the *inner loop* of a tutoring system (VanLehn, 2006). Adaptation can also occur in the *outer loop*, where the system makes or recommends decisions about what problems the student should work on next. Outer-loop adaptation can use coarse- or fine-grained models of the student's knowledge, which are typically constructed on the basis of observation of the student's behavior.

Usability Challenges

One of the reasons why the systems discussed in the first part of this chapter have been successful is that they have managed to avoid some typical usability side effects that can be caused by adaptation. These side effects were quite pronounced in some of the early user-adaptive systems that came out of research laboratories in the 1980s and 1990s, and they led to some heated discussion about the general desirability of adaptation to users (see the references given later in this section). By now, designers of user-adaptive systems have learned a good deal about how to avoid these side effects, but it is still worthwhile to bear them in mind, especially when we design new forms of adaptation that go beyond mere imitation of successful existing examples.

Figure 20.21 gives a high-level summary of many of the relevant ideas that have emerged in discussions of usability issues raised by user-adaptive systems and interactive intelligent systems more generally (see, e.g., Norman, 1994; Wexelblat & Maes, 1997; Höök, 2000; Tsandilas & schraefel, 2004; Jameson, 2009). The figure uses the metaphor of signs that give warnings and advice to persons who enter a potentially dangerous terrain.

The Usability Threats shown in the third column characterize the five most important potential side effects. A first step toward avoiding them is to understand why they can arise; 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:

[7] General sources of literature on this type of system include the *International Journal of Artificial Intelligence in Education* and the proceedings of the alternating biennial conferences on *Artificial Intelligence in Education* and on *Intelligent Tutoring Systems*. The integrative overview by VanLehn (2006) can also serve as an introduction.

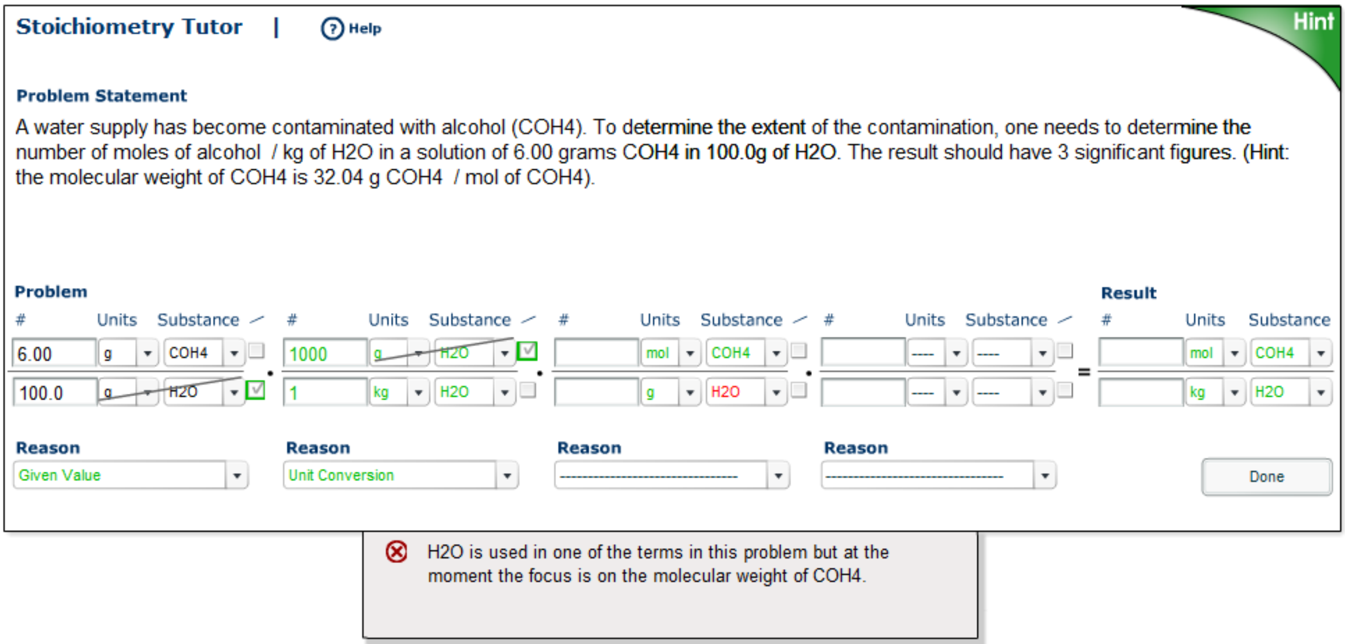


Figure 20.19: Example of error feecback provided by the STOICHIOMETRY TUTOR.

(The message below the main panel is the feedback on the student's incorrect selection of "H2O" as the "Substance" in the middle column, shown in red in the interface. Captured in February, 2011, from the tutor on http://learnlab.web.cmu.edu/ pact/chemstudy/learn/tutor2.html; reproduced with permission of Bruce McLaren.)

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. Therefore, Figure 20.21 can be used not only as a summary of some general lessons but also as a way of structuring thinking about a specific user-adaptive system; in the latter case, some of the boxes and arrows will be replaced with content that is specific to the system under consideration.

A discussion of all of the relationships indicated in Figure 20.21 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.[8] 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 two different levels of detail.

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

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, the extreme case of predictability—remaining identical over time—has the advantage that after gaining experience users may be able to engage in *automatic processing* (see, e.g., Hammond, 1987; or, for a less academic discussion, Krug, 2006): They can use the parts of the interface quickly, accurately, and with little or no attention. In this situation, even minor deviations from constancy on a fine-grained level can have the serious consequence of making automatic processing impossible or error-prone. But even a lower degree of predictability on this detailed level can be useful for the user's planning of actions. Suppose that a person who regularly visits the website for this year's CHI conference knows that, if she types "chi" into the search field of her browser, the conference's homepage will appear among the first few search results (possibly because the search is personalized and she has visited the conference page in the past): This knowledge will enable her to access the page more quickly than if the search engine's results were less predictable.

2. *Success of adaptation.* Often, all the user really needs to be able to predict and understand is the general level of success of the system's adaptation. For example, before spending time following up on a system's recommendations, the

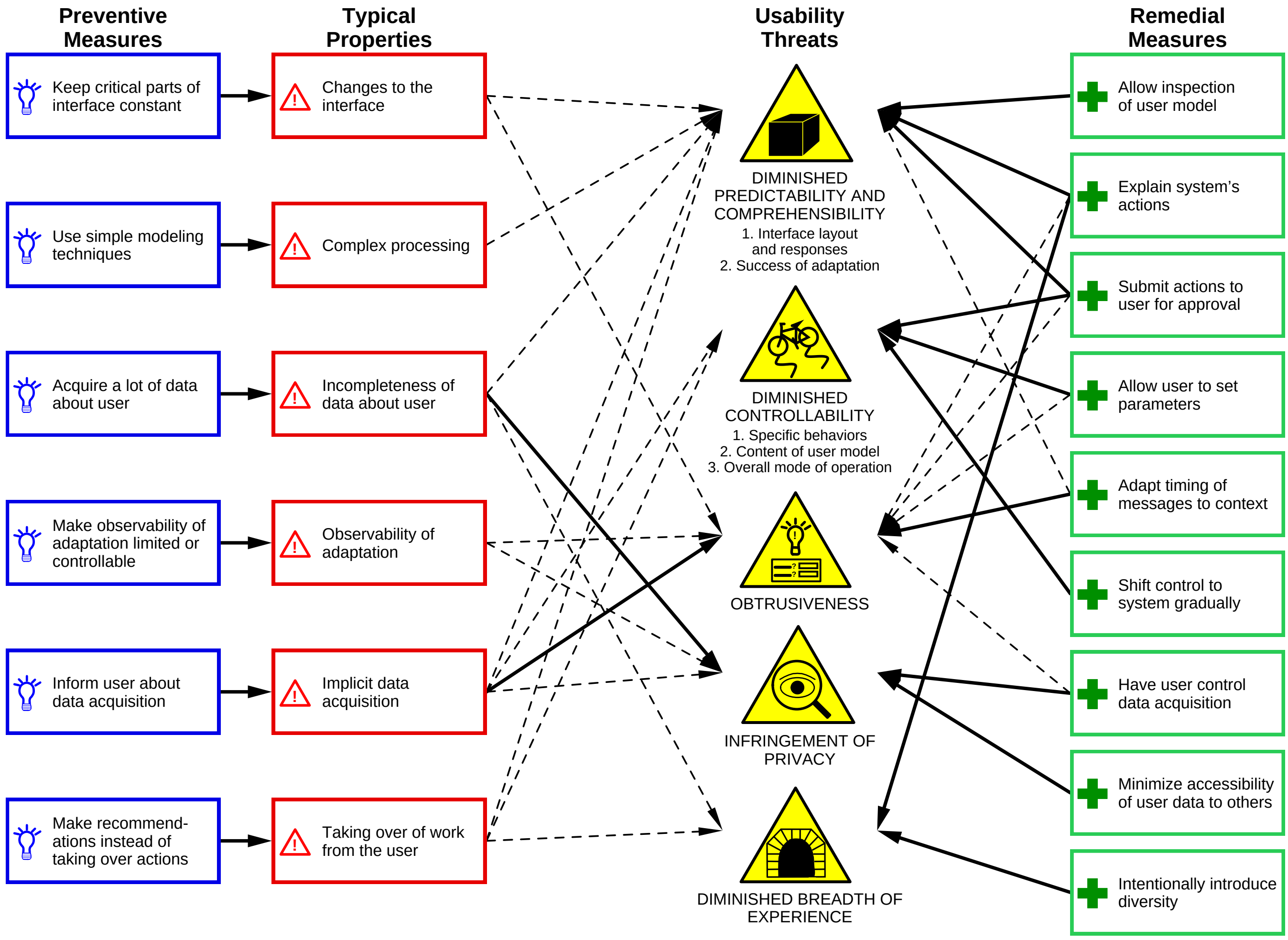


Figure 20.21: Overview of usability challenges for user-adaptive systems and of ways of dealing with them.
(Dashed arrows denote threats and solid arrows mitigation of threats, respectively; further explanation is given in the text.)

user may want to know how likely they are to be accurate. And if they turn out to be inaccurate, the user may want to understand why they weren't satisfactory in this particular case, so as to be able to judge whether it will be worthwhile to consult the recommendations in the future.

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. It is an especially important issue if the system's adaptation consists of actions that have significant consequences, such as changing the user interface or sending messages to other people. A widely used way of avoiding controllability problems is simply to have the system make recommendations, leaving it up to the user to take the actions in question. Or the system can take an action after the user has been asked to approve it. Both of these tactics can raise a threat of *obtrusiveness* (see below); so it is important to find a way of making recommendations or asking for approval in an unobtrusive fashion but still noticeable fashion (see, e.g., the discussion of AMBIENT HELP earlier in this chapter).

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. Jameson & Schwarzkopf, 2002 found that users sometimes differ strikingly in their desire for control over a given aspect of adaptation, because they attach different weight to the advantages and disadvantages of controllability, some of which are situation-specific. This observation corroborates the recommendation of Wexelblat and Maes (1997) to make 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*—were often heard in connection with early user-adaptive systems that were designed with inadequate attention to this possible side effect. Figure 20.21 shows that (a) there are several different reasons why user-adaptive systems may be obtrusive and (b) there are equally many strategies for minimizing obtrusiveness.

Threats to Privacy

Until a few years ago, threats to privacy were associated with user-adaptive systems more than with other types of system, because adaptation implied a greater need to collect and store data about individual users (see, e.g., Cranor, 2004). Nowadays, where so much of everyday life has moved to the web, people have many reasons for storing personally sensitive information (including, for example, their email, personal photos, and work documents) on computers over which they have no direct control. So the threat of privacy and security violations due to unauthorized access to or inappropriate use of personal data is now less strongly associated with the modeling of individual users. A comprehensive general discussion of privacy issues in human-computer interaction has been provided by Iachello and Hong (2007).

A privacy threat that is still specifically associated with user-adaptive systems concerns the visibility of adaptation. For example, consider a reader of GOOGLE NEWS who suffers from a particular disease and has been reading news stories related to it. If the user is not eager for everyone to know about her disease, she may take care not to be seen reading such news stories when other people are present. But if she visits the personalized section of the news site when someone else is looking and a story about the disease appears there unexpectedly, the observer may be able to infer that the user is interested in the topic: The stories that are displayed implicitly reflect the content of the user model that the system has acquired. As Figure 20.21 indicates, a preventive measure is to give the user ways of limiting the visibility of potentially sensitive adaptation.

Diminished 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 an early discussion of this issue).

Findlater and McGrenere (2010) investigated this type of tradeoff in depth in connection with personalized user interfaces that limit the number of features that a user is exposed to. Their results confirmed that this type of personalization can both increase users' performance on their main tasks and reduce their awareness of features that might be useful with other tasks. The authors discuss a number of considerations that need to be taken into account when this type of tradeoff is encountered.

As Figure 20.21 indicates, a general preventive measure is to ensure that users are free to explore the domain in question freely despite the adaptive support that the system offers. For example, recommender systems in e-commerce do not in general prevent the user from browsing or searching in product catalogs.

If a user does choose to rely heavily on the system's adaptations or recommendations, reduction of the breadth of experience is especially likely if the system relies on an incomplete user model (e.g., knowing about only a couple of the tasks that the user regularly performs or a couple of topics that she is interested in). 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 those of NETFLIX; and Linden, Hanks, & Lesh, 1997, and Shearin & Lieberman, 2001, for methods realized in different types of recommenders).

The Temporal Dimension of Usability Side Effects

The ways in which a user experiences a particular usability side effect with a given adaptive system can evolve as the user gains experience with the system. For example, adaptations that initially seem unpredictable and incomprehensible may become less so once the user has experienced them for a while. And a user may be able to learn over time how to control adaptations. In some cases, therefore, usability side effects represent an initial obstacle rather than a permanent drawback. On the other hand, since an initial obstacle may prompt the user to reject the adaptive functionality, it is worthwhile even in these cases to consider what can be done to improve the user's early experience. The remedial measure shown in Figure 20.21 of enabling the user to control the system closely at first and shift control to the system gradually is an example of such a strategy.

In general, though, the temporal evolution of the usability of an adaptive system is more complex than with nonadaptive systems, because the system tends to evolve even as the user is learning about it. A systematic way of thinking about the complex patterns that can result is offered by Jameson (2009).

Obtaining Information About Users

Any form of adaptation to an individual user presupposes that the system can acquire information about that user. Indeed, one reason for the recent increase in the prevalence of user-adaptive systems is the growth in possibilities for acquiring and exploiting such data.

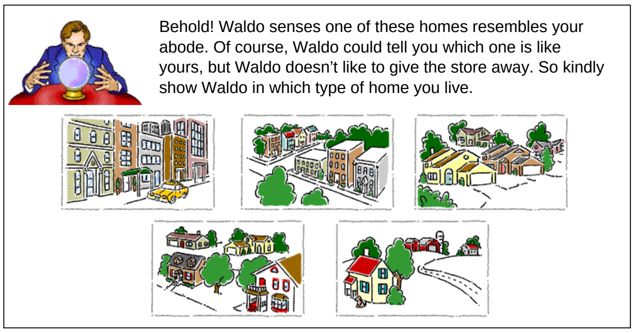


Figure 20.22: Example of a screen with which the LIFESTYLE FINDER elicits demographic information.

(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.)*

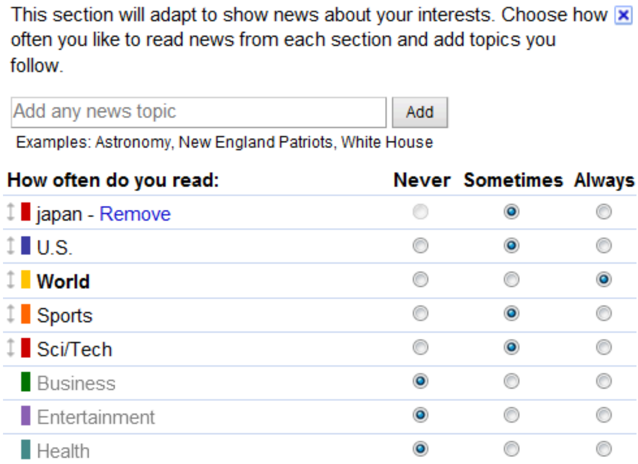


Figure 20.23: A form in which a reader of GOOGLE NEWS can characterize her interests in particular types of news.

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.

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 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. 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 early approach appeared in the web-based LIFESTYLE FINDER prototype (Figure 20.22; 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.

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 the user's interest in news items in the personalized section of GOOGLE NEWS. But it may be necessary or more efficient to ask the user for an explicit assessment. For example, shortly before this chapter went to press and after its discussion of GOOGLE NEWS had been completed, GOOGLE NEWS began providing a form (shown in Figure 20.23) on which users could specify their interests explicitly.

Because of the effort involved in this type of self-assessment and the fact that the assessments may quickly become obsolete, 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 a 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. Familiar examples include rating scales on which a user can award 1 to 5 stars and the now-ubiquitous thumbs-up "like" icon of FACEBOOK. 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. 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.

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 et al., 1997, to identify perceptual limitations relevant to the automatic generation of graphs). Trewin (2004, p. 76) 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 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).

In their pure form, naturally occurring actions require no additional investment by the user. 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 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.

Information From Social Networks

One type of information about users that has grown explosively during the last several years is information that can be found in the increasingly ubiquitous social networks (e.g., FACEBOOK, LINKEDIN, and ORKUT, but also media-sharing sites such as FLICKR). Much of this information is similar in nature to information that can in principle be found elsewhere—for example, on a user's personal homepage or in their email messages—but social networking sites encourage people to create and expose more of this information than they otherwise would. One type of information is specific to social networks: explicit links connecting people (for example, as "friends", professional collaborators, or members of the same on-line community). The most obvious way of exploiting link information was illustrated by the SOCIALBLUE system: helping people to create additional links of the same types. But the fact that a given user is a friend of another person or a member of a given community can enable the system to make many other types of inference about that user by examining the persons to whom he or she is linked (see, e.g., Brzozowski, Hogg, & Szabo, 2008; Mislove, Viswanath, Gummadi, & Druschel, 2010; Schifanella, Barrat, Cattuto, Markines, & Menczer, 2010; Zheleva & Getoor, 2009). In effect, much of the information that can be acquired in other ways summarized in this section can be propagated to other users via such links—although the nature of the inferences that can be made depends on the nature of the links and the type of information that is involved.

Other Types of Previously Stored Information

Even before the advent of social networking platforms, there were ways in which some user-adaptive systems could access relevant information about a user which was 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. 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. Related concepts are *ubiquitous user modeling* (see, e.g., Heckmann, 2005) and *cross-system personalization* (Mehta, 2009).

Low-Level Indices of Psychological States

The next two categories of information about the user started to become practically feasible in the late 1990s 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.

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), her speech (see, e.g., Liscombe, Riccardi, & Hakkani-Tür, 2005), or her eye movements (see, e.g., Conati & Merten, 2007).

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 have been diminishing steadily in importance over the years with advances in miniaturization.

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. The use of GPS (Global Positioning System) technology, often in conjunction with other signals, to determine a user's current location is familiar to most users of modern smartphones, and one of the purposes is to personalize the provision of information (e.g., about local attractions). More specialized transmitters and receivers are required, for example, if a portable museum guide 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.

Concluding Reflections

During the past few years, an increasing range of systems have been put into widespread use that exhibit some form of adaptation to users; the first two major sections of this chapter have presented a representative sample. This increasing pervasiveness can be explained in part in terms of advances that have increased the feasibility of successful adaptation to users: better ways of acquiring and processing relevant information about users and increases in computational capacity for realizing the adaptation. But there has also been a growth in understanding of the forms of adaptation that fit with the ways in which people like to use computing technology, providing added value while avoiding the potential usability side effects discussed earlier in this chapter.

One general design pattern has emerged which has been applied successfully in various forms and which might be considered the default design pattern to consider for any new form of adaptation: The nonadaptive interaction with an application is supplemented with recommendations that the user can optionally consider and follow up on.

The earliest widely used examples of this general pattern included product recommenders for e-commerce, such as Amazon.com's recommendations. As was illustrated by the examples in the first part of this chapter, the pattern has also been appearing with other functions of adaptation, such as personalized news, recommendation of people to link up with, and support for the discovery and learning of useful commands in a complex application. In tutoring systems that include an "outer loop", recommendations can concern the suggestions of learning material and exercises. Even some forms of adaptation that would not normally be called "recommendation", such as split interfaces and TASKTRACER's support for the performance of routine tasks shown in Figure 20.4, fit the same basic pattern.

The general appeal of this design pattern is understandable in that it involves making available to users some potentially helpful options which they would have had some difficulty in identifying themselves or which at least would have taken some time for them to access. This benefit is provided with little or no cost in terms of usability side effects: Provided that the available display space is adequate, the additional options can be offered in an unobtrusive way. The fact that the user is free to choose what to do with the recommended options—or to ignore them—means that any difficulty in predicting or understanding them need not cause significant problems; that the system does not take any significant action that is beyond the user's control; and that the user's experience does not have to be restricted.

This relatively straightforward and generally successful paradigm cannot be applied to all forms of adaptation to users. Adaptation to abilities and impairments often requires the provision of an alternative interface. And some types of system—such as small mobile devices, smart objects embedded in the environment, and telephone-based spoken dialog systems—may lack sufficient display space to offer additional options unobtrusively or a convenient way for users to sele ct such options. Achieving effective and widely used adaptation where the general recommendation-based design pattern cannot be applied remains a challenge for researchers and designers.

Acknowledgements

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: Jilin Chen, Thomas Dietterich, Werner Geyer, Shaun Kane, Bruce Krulwich, Justin Matejka, and Bruce McLaren. The contribution of the first author was supported in part by the 7th Framework EU Integrating Project GLOCAL: *Event-based Retrieval of Networked Media* (http://www.glocal-project.eu/) under grant agreement 248984.

References

Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education, 19*(2), 105–154.

Barnard, L., Yi, J., Jacko, J., & Sears, A. (2007). Capturing the effects of context on human performance in mobile computing systems. *Personal and Ubiquitous Computing, 11*(2), 81–96.

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).

Beales, H. (2010). *The value of behavioral targeting.* Available from (http://www.networkadvertising.org/pdfs/Beales_NAI_Study.pdf) (Study sponsored by the Network Advertising Initiative)

Bergman, E., & Johnson, E. (1995). Towards accessible human-computer interaction. *Advances in Human-Computer Interaction, 5*(1).

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.

Brettel, H., Viénot, F., & Mollon, J. D. (1997). Computerized simulation of color appearance for dichromats. *Journal of the Optical Society of America A, 14*(10), 2647–2655.

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

Brzozowski, M. J., Hogg, T., & Szabo, G. (2008). Friends and foes: Ideological social networking. In M. Burnett et al. (Eds.), *Human factors in computing systems: CHI 2008 conference proceedings* (pp. 817–820). New York: ACM.

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. (2007). Supporting interface customization using a mixed-initiative approach. In T. Lau & A. R. Puerta (Eds.), *IUI 2007: International Conference on Intelligent User Interfaces.* 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.

Canadian Marketing Association, T. (2009). *Behavioural advertising: Why consumer choice matters.*

Carmagnola, F., Vernero, F., & Grillo, P. (2009). SoNARS: a social networks-based algorithm for social recommender systems. In *Proceedings of UMAP 2009, the 17th international Conference on User Modeling, Adaptation, and Personalization.* Berlin: Springer.

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.

Chen, J., Geyer, W., Dugan, C., Muller, M., & Guy, I. (2009). "Make new friends, but keep the old" – Recommending people on social networking sites. In S. Greenberg, S. Hudson, K. Hinckley, M. R. Morris, & D. R. Olsen (Eds.), *Human factors in computing systems: CHI 2009 conference proceedings* (pp. 201–210). New York: ACM.

Chen, W.-Y., Zhang, D., & Chang, E. Y. (2008). Combinational collaborative filtering for personalized community recommendation. In *Proceedings of the Fourteenth International Conference on Knowledge Discovery and Data Mining.* New York: ACM.

Conati, C., & Merten, C. (2007). Eye-tracking for user modeling in exploratory learning environments: an empirical evaluation. *Knowledge-Based Systems*, *20*(6), 557–574.

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.

Daly, E. M., Geyer, W., & Millen, D. R. (2010). The network effects of recommending social connections. In P. Resnick, M. Zanker, X. Amatriain, & M. Torrens (Eds.), *Proceedings of the fourth ACM conference on recommender systems* (pp. 301–304). New York: ACM.

Dietterich, T. G., Bao, X., Keiser, V., & Shen, J. (2010). Machine learning methods for high level cyber situation awareness. In S. Jajodia, P. Liu, V. Swarup, & C. Wang (Eds.), *Cyber situational awareness: Issues and research.* New York: Springer.

Ehrlich, K., Lin, C.-Y., & Griffiths-Fisher, V. (2007). Searching for experts in the enterprise: Combining text and social network analysis. In *Proceedings of the 2007 International ACM Conference on Supporting Group Work* (pp. 117–126).

Federal Trade Commission, T. (2009). *Federal trade commission staff report: Self-regulatory principles for online behavioral advertising.* (Retrieved from http://www.ftc.gov/os/2009/02/P085400behavadreport.pdf in February, 2011)

Findlater, L., & Gajos, K. Z. (2009). Design space and evaluation challenges of adaptive graphical user interfaces. *AI Magazine*, *30*(4), 68–73.

Findlater, L., & McGrenere, J. (2008). Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. In M. Burnett et al. (Eds.), *Human factors in computing systems: CHI 2008 conference proceedings* (pp. 1247–1256). New York: ACM.

Findlater, L., & McGrenere, J. (2010). Beyond performance: Feature awareness in personalized interfaces. *International Journal of Human-Computer Studies*, *68*(3), 121–137.

Findlater, L., Moffatt, K., McGrenere, J., & Dawson, J. (2009). Ephemeral adaptation: The use of gradual onset to improve menu selection performance. In S. Greenberg, S. Hudson, K. Hinckley, M. R. Morris, & D. R. Olsen (Eds.), *Human factors in computing systems: CHI 2009 conference proceedings.* New York: ACM.

Gajos, K., Weld, D., & Wobbrock, J. (2010). Automatically personalized user interfaces with supple. *Artificial Intelligence*, *174*(12–13), 910–950.

Gajos, K., Wobbrock, J., & Weld, D. (2007). Automatically generating user interfaces adapted to users' motor and vision capabilities. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology* (pp. 231–240). New York, NY, USA: ACM.

Gajos, K., Wobbrock, J., & Weld, D. (2008). Improving the performance of motor-impaired users with automatically generated, ability-based interfaces. In M. Burnett et al. (Eds.), *Human factors in computing systems: CHI 2008 conference proceedings* (pp. 1257–1266). New York: ACM.

Gajos, K. Z., Czerwinski, M., Tan, D. S., & Weld, D. S. (2006). Exploring the design space for adaptive graphical user interfaces. In *Proceedings of the 2006 Conference on Advanced Visual Interfaces* (pp. 201–208).

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.

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.

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.

Heckmann, D. (2005). *Ubiquitous user modeling*. Berlin: infix.

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

Hick, W. E. (1952). On the rate of gain of information. *The Quarterly Journal of Experimental Psychology*, *4*(1), 11–26.

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.

Hwang, F., Keates, S., Langdon, P., & Clarkson, J. (2004). Mouse movements of motion-impaired users: A submovement analysis. In *Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 102–109). New York: ACM.

Iachello, G., & Hong, J. (2007). End-user privacy in human-computer interaction. *Foundations and Trends in Human-Computer Interaction*, *1*(1), 1–137.

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. (2008). Adaptive interfaces and agents. In A. Sears & J. Jacko (Eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (2nd ed., pp. 433–458). Boca Raton, FL: CRC Press.

Jameson, A. (2009). Understanding and dealing with usability side effects of intelligent processing. *AI Magazine*, *30*(4), 23–40.

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.

Jefferson, L., & Harvey, R. (2006). Accommodating color blind computer users. In *Proceedings of the Eighth International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 40–47). New York: ACM.

Jefferson, L., & Harvey, R. (2007). An interface to support color blind computer users. In B. Begole, S. Payne, E. Churchill, R. S. Amant, D. Gilmore, & M. B. Rosson (Eds.), *Human factors in computing systems: CHI 2007 conference proceedings* (pp. 1535–1538). New York: ACM.

Kane, S. K., Wobbrock, J. O., & Smith, I. E. (2008). Getting off the treadmill: Evaluating walking user interfaces for mobile devices in public spaces. In *Proceedings of the 10th International Conference on Human Computer Interaction With Mobile Devices and Services* (pp. 109–118). New York, NY: ACM.

Keates, S., Langdon, P., Clarkson, J. P., & Robinson, P. (2002). User models and user physical capability. *User Modeling and User-Adapted Interaction*, *12*(2), 139–169.

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.

Krug, S. (2006). *Don't make me think: A common-sense approach to web usability* (2nd ed.). Berkeley, CA: New Riders.

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.

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

Law, C. M., Sears, A., & Price, K. J. (2005). Issues in the categorization of disabilities for user testing. In *Proceedings of HCI International.*

Li, W., Matejka, J., Grossman, T., Konstan, J., & Fitzmaurice, G. (2011). Design and evaluation of a command recommendation system for software applications. *ACM Transactions on Computer-Human Interaction*, *18*(2).

Lin, M., Goldman, R., Price, K. J., Sears, A., & Jacko, J. (2007). How do people tap when walking? An empirical investigation of nomadic data entry. *International Journal of Human-Computer Studies*, *65*(9), 759–769.

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.

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.

Liu, J., Dolan, P., & Pedersen, E. R. (2010). Personalized news recommendation based on click behavior. In M. O. Cavazza & M. X. Zhou (Eds.), *IUI 2010: International Conference on Intelligent User Interfaces* (pp. 31–40). New York: ACM.

Mackay, W. E. (1991). Triggers and barriers to customizing software. In S. P. Robertson, G. M. Olson, & J. S. Olson (Eds.), *Human factors in computing systems: CHI 1991 conference proceedings* (pp. 153–160). New York: ACM.

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

Matejka, J., Grossman, T., & Fitzmaurice, G. (2011). Ambient Help. In D. Tan, B. Begole, W. Kellogg, G. Fitzpatrick, & C. Gutwin (Eds.), *Human factors in computing systems: CHI 2011 conference proceedings.* New York: ACM.

Matejka, J., Li, W., Grossman, T., & Fitzmaurice, G. (2009). CommunityCommands: Command recommendations for software applications. In *UIST 2009 Conference Proceedings: ACM Symposium on User Interface Software and Technology* (pp. 193–202).

McGrenere, J., Baecker, R. M., & Booth, K. S. (2007). A field evaluation of an adaptable two-interface design for feature-rich software. *ACM Transactions on Computer-Human Interaction*, *14*(1).

McLaren, B. M., DeLeeuw, K. E., & Mayer, R. E. (2011a). A politeness effect in learning with web-based intelligent tutors. *International Journal of Human-Computer Studies*, *69*, 70–79.

McLaren, B. M., DeLeeuw, K. E., & Mayer, R. E. (2011b). Polite web-based intelligent tutors: Can they improve learning in classrooms? *Computers & Education*, *56*, 574–584.

Mehta, B. (2009). *Cross system personalization: Enabling personalization across multiple systems*. Saarbrücken, Germany: VDM Verlag.

Mislove, A., Viswanath, B., Gummadi, P. K., & Druschel, P. (2010). You are who you know: Inferring user profiles in online social networks. In *Proceedings of the Third International Conference on Web Search and Web Data Mining* (pp. 251–260). New York, NY, USA.

Mitchell, J., & Shneiderman, B. (1989). Dynamic versus static menus: An exploratory comparison. *SIGCHI Bulletin*, *20*(4), 33–37.

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.

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

Palen, L. (1999). Social, individual and technological issues for groupware calendar systems. In M. G. Williams, M. W. Altom, K. Ehrlich, & W. Newman (Eds.), *Human factors in computing systems: CHI 1999 conference proceedings* (pp. 17–24). New York: ACM.

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

Pu, P., & Chen, L. (2008). User-involved preference elicitation for product search and recommender systems. *AI Magazine*, *29*(4), 93–103.

Rhodes, B. J. (2000). *Just-in-time information retrieval*. Unpublished doctoral dissertation, School of Architecture and Planning, Massachusetts Institute of Technology, Cambridge, MA.

Rich, C. (2009). Building task-based user interfaces with ANSI/CEA-2018. *IEEE Computer*, *42*(8), 20–27.

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).

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.

Schifanella, R., Barrat, A., Cattuto, C., Markines, B., & Menczer, F. (2010). Folks in folksonomies: Social link prediction from shared metadata. In *Proceedings of the Third International Conference on Web Search and Web Data Mining* (pp. 271–280). New York, NY, USA.

Shami, N. S., Yuan, Y. C., Cosley, D., Xia, L., & Gay, G. (2007). That's what friends are for: Facilitating 'who knows what' across group boundaries. In *Proceedings of the 2007 International ACM Conference on Supporting Group Work* (pp. 379–382).

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.

Shen, J., Fitzhenry, E., & Dietterich, T. (2009). Discovering frequent work procedures from resource connections. In N. Oliver & D. Weld (Eds.), *IUI 2009: International Conference on Intelligent User Interfaces.* New York: ACM.

Shen, J., Irvine, J., Bao, X., Goodman, M., Kolibab, S., Tran, A., et al. (2009). Detecting and correcting user activity switches: Algorithms and interfaces. In N. Oliver & D. Weld (Eds.), *IUI 2009: International Conference on Intelligent User Interfaces.* New York: ACM.

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.

Teevan, J., Dumais, S. T., & Horvitz, E. (2010). Potential for personalization. *ACM Transactions on Computer-Human Interaction*, *17*(1).

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

Tintarev, N., & Masthoff, J. (2010). Explanation of recommendations. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook.* Berlin: Springer.

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. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence and Education*, *16*(3), 227–265.

Vasuki, V., Natarajan, N., Lu, Z., & Dhillon, I. S. (2010). Affiliation recommendation using auxiliary networks. In P. Resnick, M. Zanker, X. Amatriain, & M. Torrens (Eds.), *Proceedings of the fourth ACM conference on recommender systems.* New York: ACM.

Vig, J., Sen, S., & Riedl, J. (2009). Tagsplanations: Explaining recommendations using tags. In N. Oliver & D. Weld (Eds.), *IUI 2009: International Conference on Intelligent User Interfaces* (pp. 47–56). 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, cited with permission.)

Wobbrock, J. O., Kane, S. K., Gajos, K. Z., Harada, S., & Froelich, J. (2011). Ability-based design: Concept, principles and examples. *ACM Transactions on Accessible Computing*.

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.

Zheleva, E., & Getoor, L. (2009). To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th International Conference on the World Wide Web* (pp. 531–540). Madrid, Spain.

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).