

Understanding and Dealing With Usability Side Effects of Intelligent Processing

Anthony Jameson

■ *The inclusion of AI in an interactive system brings many potential benefits, but there are also potential side effects for the usability of the systems that ought to be taken into account from the very start. This article offers a systematic method for analyzing such usability side effects, the goal being to provide a solid basis for decisions about how to avoid or mitigate them. The analysis schema is applied in turn to nine classes of side effect. Many ideas that have been discussed in earlier literature are synthesized within this framework, which also brings into focus some concepts and issues that have received little attention so far.*

There are powerful arguments to the effect that the inclusion of AI in interactive systems can make them more powerful and easier or more natural to use. The historical overview in this issue by Grudin (2009) reports on some arguments to this effect over the past few decades. The section “Taming the Savage Beast of HCI” in the theme article by Lieberman (2009) offers a recent perspective.

But an argument has been voiced about equally often (see the references given below) that makes a contrasting claim: that the introduction of intelligence into an interactive system can cause usability problems that may in fact outweigh any benefits. These unintended negative consequences of the introduction of intelligence often have no direct relationship with the intended benefits, just as the adverse effects of a medication may bear no obvious relationship to the intended benefits of taking that medicine. Therefore, these negative consequences can be seen as *side effects*.

The purpose of this article is to give designers, developers, and users of interactive intelligent systems a detailed awareness of the potential side effects of AI. As with medications, awareness of the side effects can have different implications: We may be relieved to see that a given side effect is unlikely to occur in our particular case. We may become convinced that it will inevitably occur and therefore decide not to “take the medicine” (that is, decide to stick with mainstream systems). Or most likely and most constructively, by looking carefully at the causes of the side effects and the conditions under which they can occur, we can figure out how to exploit the benefits of AI in interactive systems while avoiding the side effects.

This article has drawn inspiration from many previous essays on this general topic, including the following: Norman (1994),

Lanier (1995), Horvitz (1999), Höök (2000), Tsandilas and Schraefel (2004), Jameson (2008).¹ Because many points have been discussed (in different terms) by different authors, it would be pedantic to provide citations for each idea that has been mentioned in the literature.

The primary novel contribution of this article can be summarized as follows: A general schema is introduced for describing usability side effects that explicitly represents their causes, their consequences, the conditions under which these consequences are likely to occur, and typical changes in these consequences over time. Application of this schema to a given class of side effects not only increases researchers' understanding of them but also suggests strategies for preventing and mitigating them. This style of analysis introduces more structure than is found in earlier discussions, and it brings into focus some concepts and issues that have so far attracted little attention—in particular those that concern the evolution over time of the side effects and of users' responses to them.

Basic Schema for the Discussion of Usability Side Effects

The basic schema is shown abstractly in figure 1. Each analysis will consider the following aspects of a given class of side effect: causes, consequences, changes over time, and prevention.

1. Causes: What are the typical characteristics of interactive intelligent systems that give rise to this type of side effect, and how can they be manifested in the system's interaction with the user? In general, the appearance in the system's behavior is not inevitable, so the diagram indicates conditions under which it is likely. Understanding the causes helps us to see when the side effect is likely to occur—and it also suggests ways of preventing it from occurring.

2. Consequences: What are the possible consequences for the user? We speak of the user's *responses* to refer either to aspects of the user's behavior that are influenced by the system or to subjective reactions like frustration. Here again, the possible responses are not inevitable, so the diagram usually indicates one or more conditions for them.

3. Changes over Time: How do these side effects tend to change as a user acquires more experience with a given system?

With most side effects, there is some development over time, which may be in the direction of aggravation of the side effect or (more commonly) tapering off over time. It is therefore useful to distinguish roughly between early use of the system and later use. Except in the case of systems that people use for only a brief time and never again (for example, an interactive exhibit in a museum), designers are interested in ensuring that both ear-

ly and later use are reasonably satisfactory. In particular, even if "later use" will ultimately constitute 98 percent of all of the use, if early use is not satisfactory, users may abandon the system before ever reaching the stage of later use.

With mainstream systems, the evolution from early to later use consists mainly in changes in the user's knowledge and skill. With intelligent systems, there is often also change in the system itself, which may acquire knowledge and adapt to the individual user. The transition between early and late use may therefore be the result of a complex combination of changes in the user and the system.

4. Prevention (not shown in figure 1): What are some general strategies for preventing the side effect from occurring—or for mitigating its consequences?

One of the benefits of this type of analysis is that it makes possible a systematic approach to the generation of ways of preventing or mitigating side effects. In particular, each of the arrows in the diagrams that specifies a condition suggests a way of combatting the side effect in question: by trying to ensure that the condition does not hold. We will see many examples of this general approach. (Because there are usually many possible prevention strategies, these are not shown directly in the diagrams.)

There are also many specific strategies and guidelines that apply to particular types of system; many of these can be viewed as special instances of the high-level strategies mentioned here. The case study articles in this special issue offer a number of examples of such strategies.

To make it easier for the reader to get accustomed to the concepts just introduced, we will discuss the nine classes of side effects in order of increasing complexity, one in each of the following sections: Need to Switch Applications or Devices, Need to Teach the System, Narrowing of Experience, Unsatisfactory Aesthetics or Timing, Need for Learning by the User, Inadequate Control over Interaction Style, Threats to Privacy, Inadequate Predictability and Comprehensibility, and Imperfect System Performance.

The first, fourth, and fifth of these categories are new in this article; the other classes have been discussed in previous literature, though with different definitions of the classes.

Need to Switch Applications or Devices

Sometimes, useful intelligent technology can for one reason or another not be realized within any of the applications, or with the same hardware, that users are accustomed to working with. Users may therefore be forced to switch between their usual applications and hardware and something

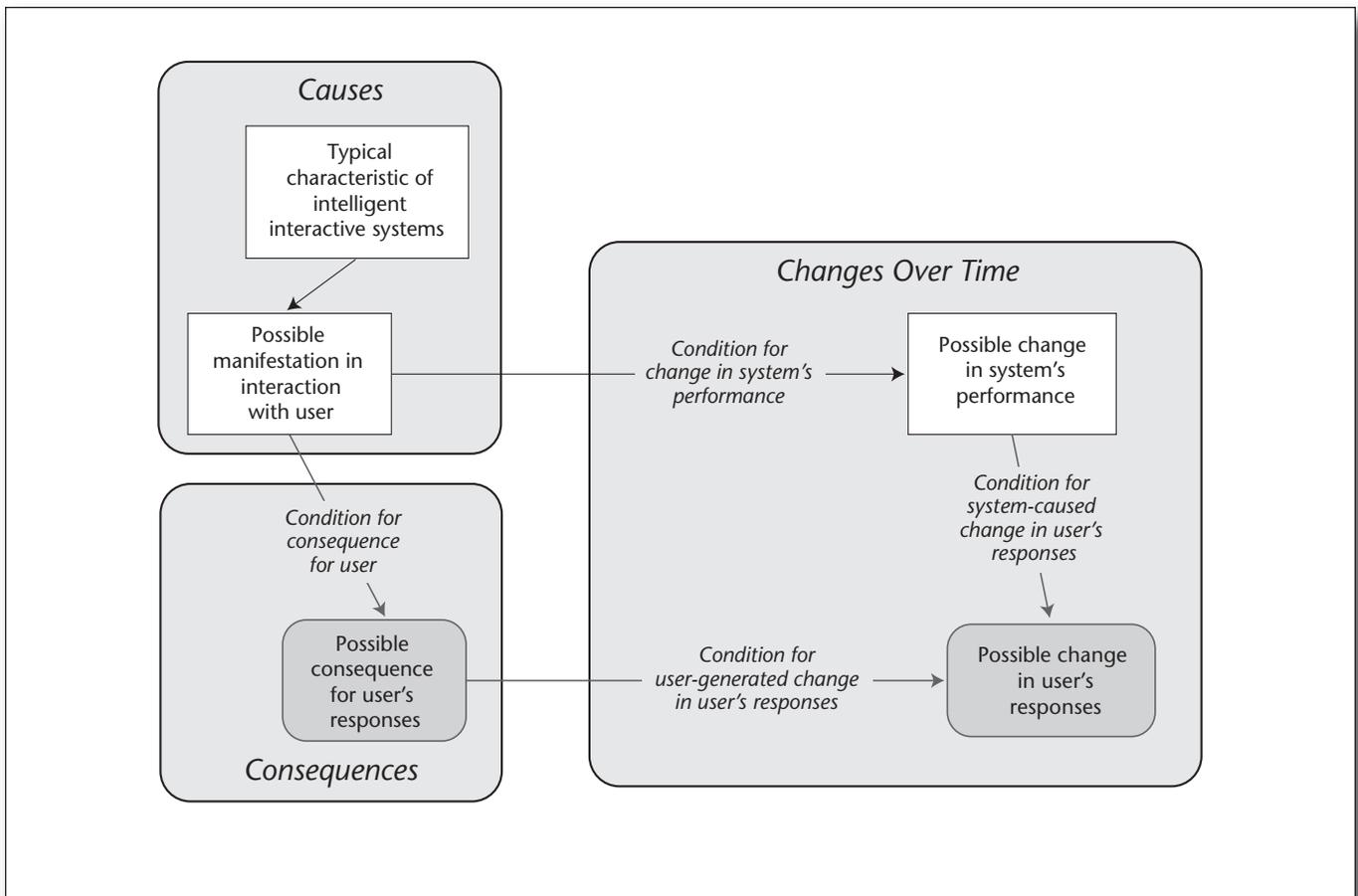


Figure 1. Schema for Analyzing Usability Side Effects.

Rectangles: properties of system; rounded boxes: responses of user.

more specialized in order to take advantage of the system's intelligence. (The analysis of this type of side effect is shown in figure 2.)

Causes

Sometimes, intelligent technology places demands on hardware that are not met by the computing environments that users are accustomed to working with. For example, unusual input and output modalities may require nonstandard input or output devices. There may also be more mundane reasons that are not specific to intelligent processing: for example, because of the general policy for the Apple iPhone that no systemwide alternative to the usual text input methods could be installed, the innovative ShapeWriter text input method described in the case study by Kristensson (2009) could not be introduced as a general alternative input method for the iPhone but rather only as part of a separate application.

Similarly, the SMARTedit system described in the case study by Lau (2009) offered powerful support to the user in performing text editing operations,

but it did not work within a normal text editor; so the user had to switch to a different system in order to perform text editing operations that SMARTedit could help with.

Consequences

The immediate consequence is that the user needs to invest more time and effort in the use of the intelligent system.

In general, however, users do not simply put up with a negative side effect; instead they adapt their behavior somehow, as we will see later in more complex cases. A natural adaptation here is to use the functionality in question only in cases where the benefits seem especially great; after all, they need to be great enough to justify the effort of switching. For example, Lau (2009) reports that people tended to use SMARTedit only for complex tasks for which it seemed to promise a large benefit.

Changes over Time

The need to perform this type of switching will in general not diminish over time, but there may be

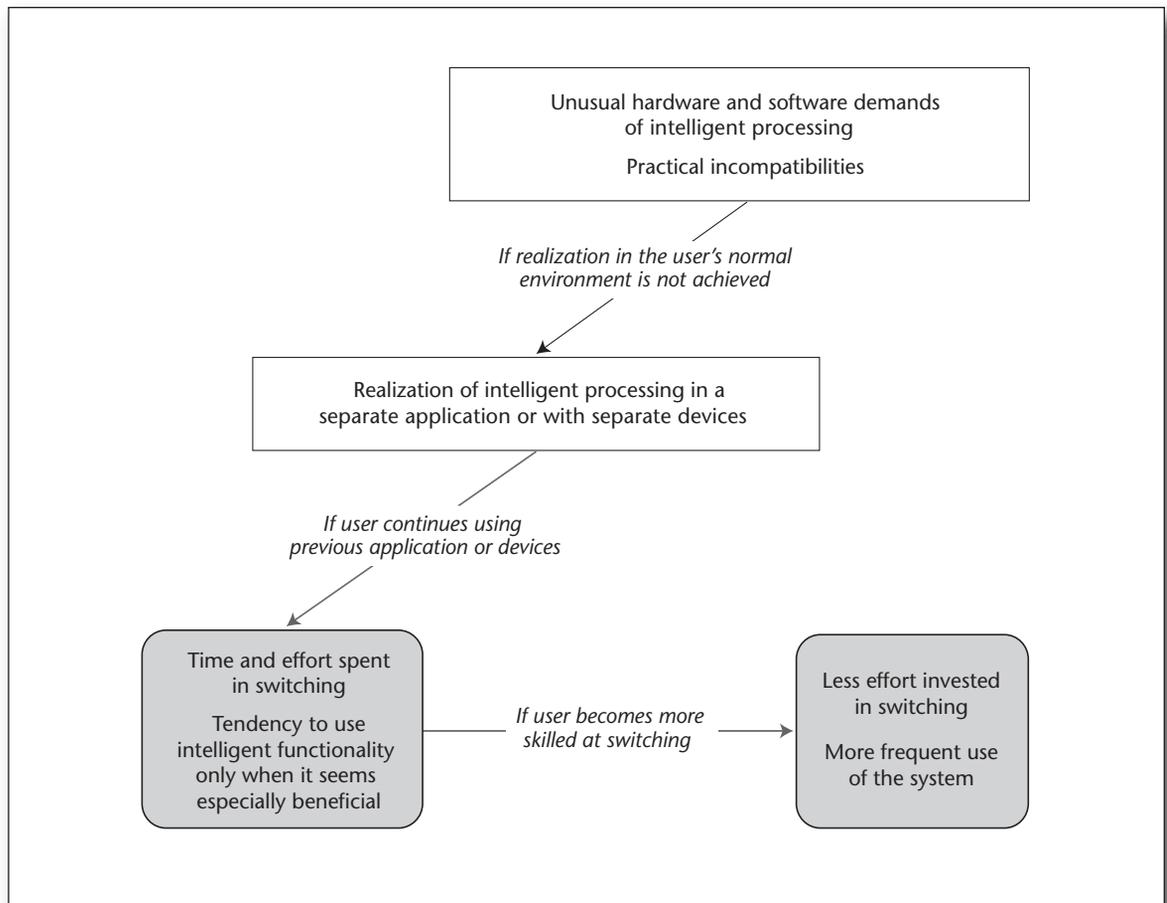


Figure 2. Analysis of Usability Side Effects in the Category Need to Switch Applications or Devices.

opportunities for the user to become more skilled at it. For example, with some trial and error, a user might learn the fastest possible procedure for composing an SMS message using ShapeWriter and transferring it to a separate messaging application. If this sort of skill improvement does occur, and if the user notices it, one consequence may be the more frequent use of the new system, because of the decline in the cost of using it.

Prevention

The obvious high-level strategy is to try to minimize the effort associated with switching by fitting the intelligent system as smoothly as possible into the target users' normal work environment or by at least making it clear to users how they can do the switching as easily as possible. The danger should be borne in mind that, if the cost of switching for early users is too high, they may use the system so infrequently that they never become skilled at doing the switching and may never acquire an adequate appreciation of the benefits that the system can bring (this type of problem is discussed more generally in the section Need for Learning by the User).

Need to Teach the System

Some types of intelligent interactive system require the user to spend some time, at least during the early phase of use, providing information that the system requires in order to apply its intelligent processing (figure 3).

Causes

Intelligent systems tend to make use of knowledge of various sorts about the tasks that they perform and the users for whom they are performing them. Though they may be able to acquire much of this knowledge automatically, often some sort of teaching by the user is required—for example, the provision of information about the tasks that the user needs help with. Systems whose intelligence consists in their ability to adapt to the user may require some explicit input about the user's abilities, tastes, or whatever other properties are relevant to the type of adaptation that is involved.

Consequences

An immediate consequence, if the user is inclined to provide the requested information, is an invest-

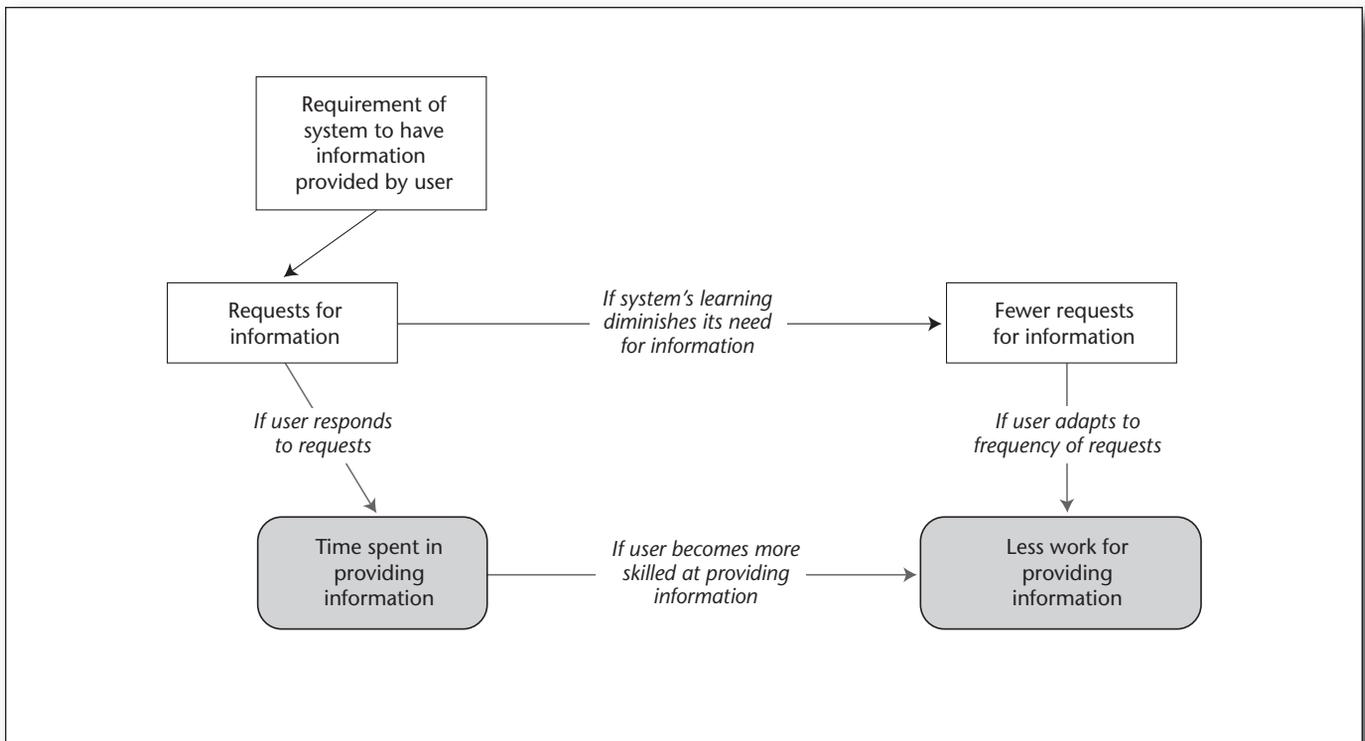


Figure 3. Analysis of Usability Side Effects in the Category Need to Teach the System.

ment of time and attention by the user, which may be viewed as a significant cost associated with the use of the system.

Changes over Time

There are two reasons why this side effect may diminish in importance over time: If the system manages to learn cumulatively, it may require less and less new information. Also, the task of supplying information to the system is one that the user may become more skilled at, so that the costs decline.

Even in this favorable case, though, there remains the problem of getting the user through the early phase in which using the system has considerable costs and the user may not yet have enough experience with the benefits to be convinced that the investment is worthwhile.

Prevention

One theoretically possible strategy with regard to the phase of early use is to make it clear to users (for example, by showing an example of a contented experienced user) how great the ultimate benefits will be if they manage to put up with the initial demands placed by the system. This strategy is comparable to that of persuading young people to invest in a retirement fund by making it clear to them how much better off they will be once they reach retirement age. One problem in

both cases is that people tend to discount future benefits (see, for example, Weber and Johnson [2009]); and in fact they may be justified in doing so if they do not (yet) have a clear picture of how these benefits will look for them and cannot be certain that they will really be delivered. (Note that young people who, in early 2008, failed to put money away in a stock-based retirement fund can in retrospect be seen to have made the right choice.)

Not surprisingly, then, researchers often devote considerable attention to ways of minimizing the initial demands of an intelligent system, for example by looking for machine-learning methods that can do something useful after seeing only a handful of training examples. Even if a method like this is less effective in the long run than an alternative method, it may be preferable on the grounds that users would be unlikely to stay around long enough to enjoy the benefits of the more effective method anyway.

Narrowing of Experience

Interactive intelligent systems may lead to a narrowing of the user's knowledge or experience, relative to the situation in which the user works with more conventional software (figure 4).

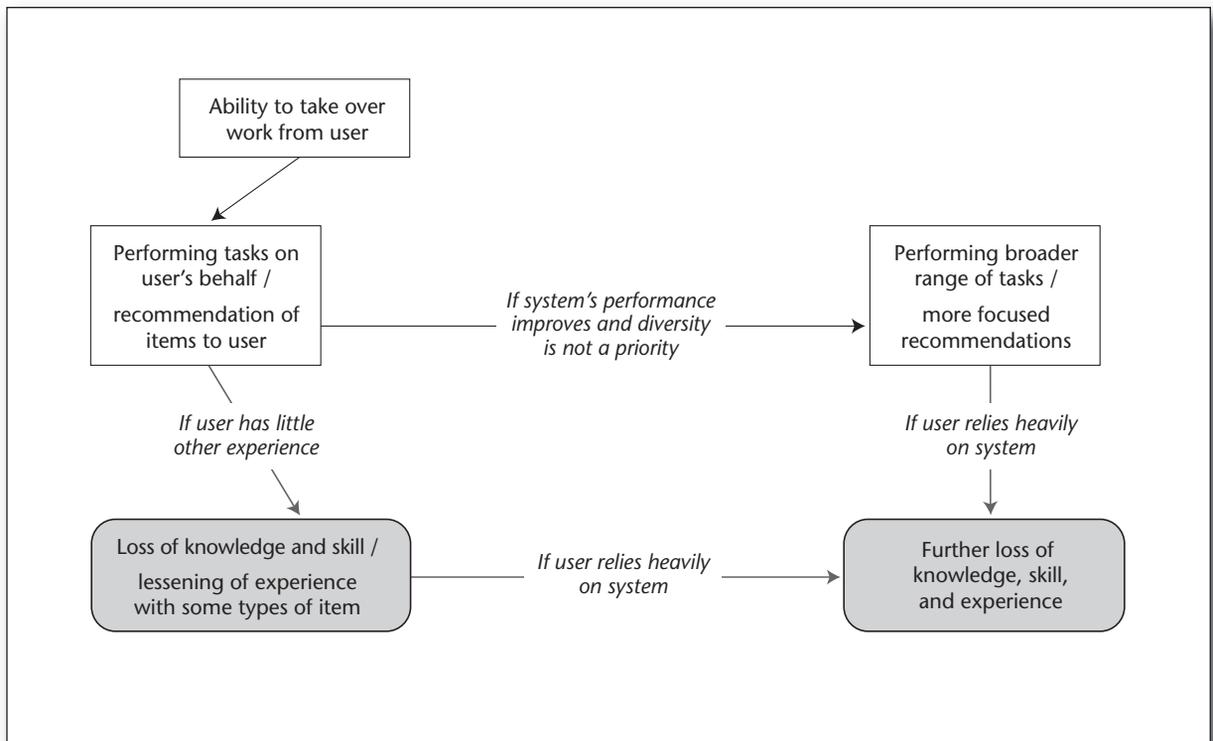


Figure 4. Analysis of Usability Side Effects in the Category Narrowing of Experience.

Causes

One of the general strengths of intelligent systems is their ability to take over from the user work that users would normally have to do themselves, even if this work requires some capabilities that are normally associated with human intelligence (for example, in the case study of Faulring et al. [2009], the identification of tasks implied by e-mail messages).

Some interactive intelligent systems do not take a task entirely off of the users' hands but rather recommend to users particular items that they would normally select on their own, such as movies or products. In an effort to recommend items that the user is especially likely to appreciate, a system may tend to recommend things that are similar to things that the user already knows and likes.

Consequences

The consequences associated with the first case are those that are typical of delegation when it is done among humans as well: the person who delegates becomes less skilled at the tasks in question. This consequence is not necessarily a bad thing. As Maes (Shneiderman and Maes 1997) put the point: the person who delegates may be happy not to have to learn about the task in question because he or she prefers to learn about other things.

In the case of recommendations, if the user relies heavily on such recommendations and does not

make use of other means of identifying items to consider, the range of items that the user experiences may become narrower than is good for the user. But here again, some narrowing of experience may be considered a good thing. Presumably few e-mail users regret the way that their intelligent spam filter prevents them from learning about the full range of performance-enhancing drugs and Nigerian business opportunities.

Changes over Time

In contrast to most of the other usability side effects that we consider, narrowing of experience tends not to be a problem during early use but rather to become an increasing problem over time. When the user encounters the potentially experience-narrowing system for the first time, the initial encounter may actually lead to a broadening of experience in the short term. Note also that the system may be able to take over a larger proportion of the user's tasks as it learns more; and depending on the recommendation method used, a recommender may offer more and more narrowly focused recommendations.

Prevention

Given that, as was just discussed, it should be left up to users to decide to what extent their experience should be narrowed, the goal of prevention is not so much to avoid narrowing at all cost but

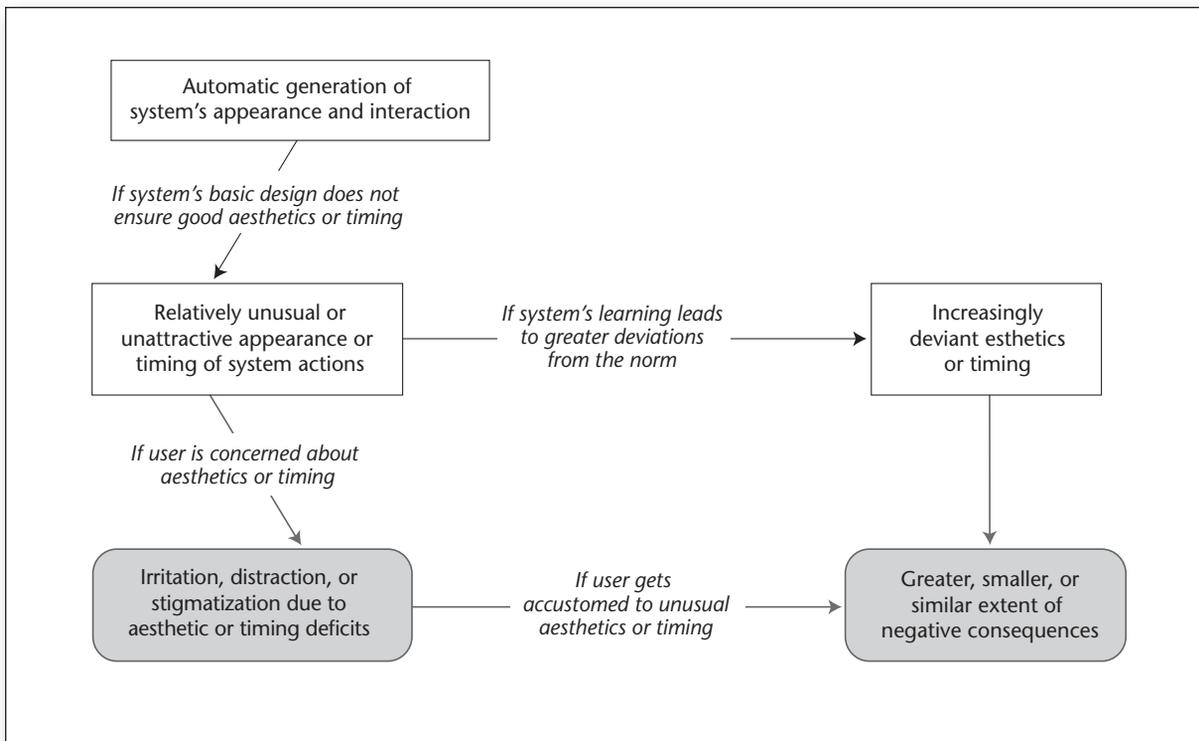


Figure 5. Analysis of Usability Side Effects in the Category Unsatisfactory Aesthetics or Timing.

rather to make it possible for people to use the system without having their experience narrowed more than they would like.

With regard to the first type of narrowing, logically implied strategies include the following:

First, the system might users to get involved in the performance of the tasks to some extent if they like, for example, by obtaining information about the task as it is being performed and making some of the decisions.

Another variant might be to supply post hoc reports on task performance, to give the user at least a vicarious experience of performing the tasks.

Second, in the case of recommendation, one approach that has been explored in recent years is to develop recommendation algorithms that assign some priority to diversity as well as to accuracy (see, for example, Ziegler et al. [2005]).

Aiming to support diversity may not even be necessary if the nature of the recommender system ensures that users will always have access to other sources of information (for example, the full product catalog on an e-commerce website) that can compensate for any narrowness of the system. In this case, the main goal is to ensure that the system being developed should not make it unnecessarily difficult to make use of these other sources of information.

A more philosophical question arises in cases

where a new system makes it possible for a user to become narrow-minded even if the user could easily enough avoid this narrowness. In some contexts, such as educational ones, it could be argued that it is the responsibility of the system designer to ensure that users do not succumb to the temptation to allow their experience to be narrowed. In this case, it may be necessary to apply more aggressive strategies than those just discussed to avoid narrowness.

Unsatisfactory Aesthetics or Timing

Interfaces to interactive intelligent systems are sometimes less aesthetically pleasing and less reassuringly familiar than interfaces to more conventional systems—in terms of their appearance or the timing of their behavior (figure 5).

Causes

With systems that do not embody intelligence, the appearance of the interface is typically determined by a skilled and talented interface designer who aims, among other goals, to create a user experience that is aesthetically pleasing and that fits well into the user's experience with other applications and devices.

Some problems with aesthetics can be seen even in mainstream interfaces when the designer loses some control over the exact appearance of the

interface—for example, when the user is given the opportunity to adjust interface elements like colors and font sizes. Interfaces to interactive intelligent systems can allow an even lower degree of control by the interface designer, in cases where the system's appearance is automatically generated or adapted by the system. For example, the interfaces for users with motor impairments that were generated by the SUPPLE system (Gajos, Wobbrock, and Weld [2008], mentioned in the case study by Findlater and Gajos [2009]), were intended to maximize the efficiency of users given their particular combination of motor impairments; aesthetics and conventional appearance were not assigned high priority. The automatically generated interfaces were in fact found to be aesthetically less appealing by some of the users studied. (These were mainly the users who had no impairments; those with impairments, who stood to gain more by working with the unusual interfaces, interestingly tended to see no aesthetic difference.)

In addition to the appearance of an interface, the timing of its behavior is an aspect that can be found disturbing by users if it is not designed carefully enough. Issues of timing have attracted attention with mainstream interfaces as well, in connection with issues such as the timing of notifications about the arrival of messages. Intelligent systems introduce the additional phenomena of (1) proactivity, where a system autonomously generates an action, proposal, or interface adaptation that the user may need to deal with; and (2) requests for input (for example, confirmation) that aim to give the user some degree of control over the system's processing.

Consequences

The most obvious consequence of deviations from conventional aesthetic expectations is that the user finds it less attractive to use the system. A less obvious consequence lies in the fact that unusual or unattractive interfaces may be seen as announcing the fact that the user has special needs that prevent him or her from using the same interfaces as everyone else (see the discussion of privacy side effects).

With regard to the timing of system actions, a frequently discussed and studied consequence is distraction: if a system calls for the user's attention at a time when the user should be concentrating on something else, it is likely that the user will at least be irritated by the distraction; users may also devote inadequate attention to either their original task or the system's communication. In both of these cases there is no limit to how negative the consequences can be, the most frequently discussed examples being those of traffic accidents caused by poorly timed system actions.

Changes over Time

There are three independent tendencies that can be at work:

First, assume that there is no evolution over time of the system's appearance and behavior. For example, the automatically generated interfaces of Gajos, Wobbrock, and Weld (2008) were created in one pass and presented to the users, who worked with them without any further adaptation of the interface. In this case, the side effect itself does not get worse over time, and the user has a chance to become accustomed to the unusual aesthetics or timing. Even the problem of distraction by poorly timed system actions may diminish with experience, as the user learns how much attention to pay to various types of system action in various situations.

The evolution over time is less favorable, however, if the system's appearance becomes more unusual and less aesthetically pleasing over time, because of incremental adaptation. For example, the adaptive interfaces of Gajos, Wobbrock, and Weld (2008) might be redesigned in an incremental way so that the user starts with a conventional interface, which is gradually adapted as the system learns about the user's impairments. This approach would at least have the advantage that the user would not initially be confronted with any problem of aesthetics or familiarity.

Third, the system may learn over time how to adjust its appearance and dynamic behavior to the user's preferences and situation; and such adaptation should tend to diminish this type of side effect.

In sum, the time course of this type of side effect can differ from one system to the next, depending on how these various possible tendencies work out.

Prevention

Several strategies for prevention of these side effects are suggested by this discussion.

With regard to aesthetics and familiarity, it can help if designers consider from the start the importance of these factors and make early design decisions that take them into account. For example, the designer can choose a basic appearance that resembles aesthetically pleasing, familiar interfaces and ensure that automatic generation and adaptation cannot lead to large deviations from the basic pattern. For instance, the task-centered e-mail processing interface of the RADAR system (described in the case study of Faulring et al. [2009]) bears a superficial resemblance to many conventional e-mail clients, and the system's intelligent processing does not result in changes to the interface's basic appearance.

If it is necessary to allow the system to alter the interface's appearance drastically, the goals of the automatic adaptation can at least include aesthetics, along with the more obvious goals like effi-

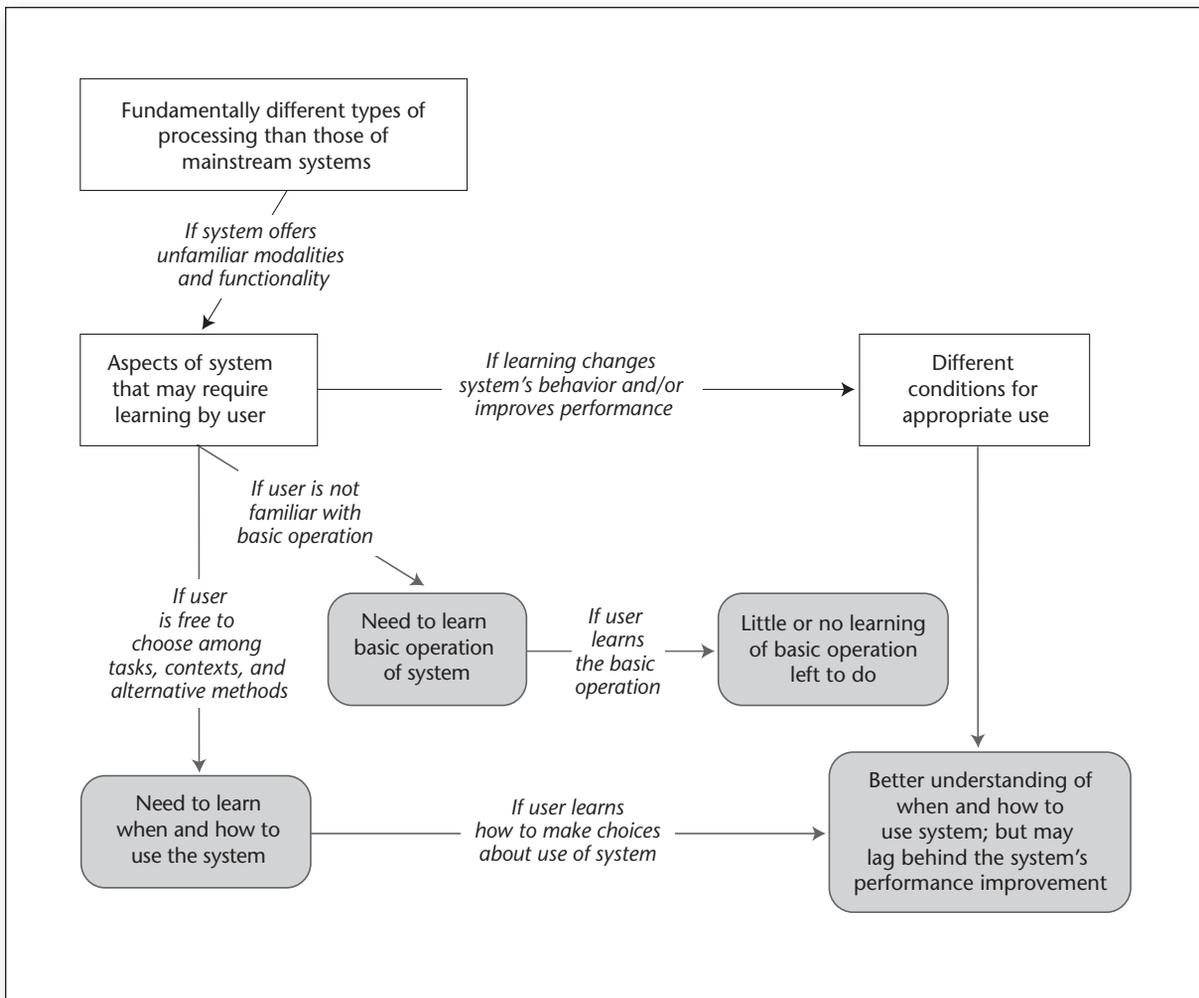


Figure 6. Analysis of Usability Side Effects in the Category Need for Learning by the User.

ciency. Consistently realizing this strategy may require additional research on the automatic generation of aesthetically pleasing interfaces.

Our analysis of changes over time suggests that having the system's appearance evolve slowly at first may have the advantage of increasing acceptability during the critical early stage of use, though there is also the danger that the benefits of the system's intelligence may not become evident as quickly.

With regard to issues of timing, a good deal of recent research has addressed the question of how the timing of the system's actions can best be adapted to the activities and preferences of the user (see, for example, Iqbal and Bailey [2006] and the on-line bibliography at interruptions.net).

Need for Learning by the User

Users may need to acquire some knowledge and experience before they can make effective use of the intelligence offered by a system (figure 6).

Causes

One of the motivations for incorporating AI in interactive systems is to reduce the need for certain types of learning on the part of the user. But as a side effect, sometimes other types of learning are necessary if the user is to be able to exploit the system's intelligence.

As was mentioned in the discussion of the need to switch applications or devices, intelligent interactive systems tend to be different from the applications that people deal with most of the time; and these differences may require learning. For speech recognition, the user needs to learn something about suitable microphones and their correct positioning and how to speak in a way that facilitates accurate recognition. Similar points apply to other unfamiliar input methods. The output of an interactive intelligent system may take unfamiliar forms, such as innovative visualizations or the autonomous performance by the system of tasks that the user would normally perform.

A less obvious form of learning concerns the

need to learn in what situations and in what ways it makes sense to use the system. Consider, for example, a smartphone that allows you to dictate e-mail messages while on the go: Does it make sense to use this text entry method only while you're on the move and unable to use the smartphone's small keyboard; or does it make sense to use it even when you are sitting still? Does the speech input work well only for messages that use simple, common language? How socially acceptable is it to dictate messages when other people are nearby? It may take users a long time to figure out the answers to such questions even after they have mastered the basic operation of the system. The difficulty is especially great for intelligent systems, because it tends to be hard to predict how they will work in novel situations, because of their general unfamiliarity and the fact that the level of their performance can depend on a variety of factors.

Consequences

When it comes to learning the basic operation of an intelligent system, the learning process may not be much different than it is for mainstream systems. Still, this learning is an aspect of system use that needs to be taken into account when we anticipate how users will respond to a new system.

The consequences of not knowing when and in what way a system can most profitably be used are more subtle. If a person regularly uses a system in situations for which it is not well suited, the problems that arise may be attributed to weaknesses in the system rather than to a poor choice of situations in which to use it. If our example smartphone user tries dictating an e-mail message that contains a lot of proper names and technical jargon, that user may conclude that the system is fundamentally useless, not that he or she should use it for dictating other types of texts.

Changes over Time

Learning the basic operation of a new interactive intelligent system tends to be most demanding at the time of the initial encounter with the system and to diminish quickly in importance. Many of the things that need to be learned can be mastered quickly, and the number is not infinite.

The process of learning when and how it makes sense to use the system can in principle go on indefinitely, because of the wide variety of situations and tasks that can be encountered and the number of factors that determine the appropriateness of a particular way of using the system. To complicate matters further, suppose that the system itself does not remain constant but rather improves its performance over time. This development is in itself positive, but it will bring relatively few benefits if the user's understanding of when and how to use the system does not keep pace with

the improvements in the system. For example, maybe after a long period of system adaptation it is feasible for the user to dictate an e-mail containing highly technical language; if the user tried doing so early on and noticed that it didn't work well, he or she might never try it again.

Once again, we see a situation where the balance of benefits and costs is relatively unfavorable during early use, representing an obstacle that users need to be helped to overcome if they are to reach a point where the use of the system is clearly worthwhile.

Prevention

The obvious high-level strategy with regard to the user's learning of the basic operation of the system is to consider exactly what these easily overlooked learning requirements are and how they can be minimized.

Strategies for helping users to figure out when and how to use an intelligent system are a relatively new topic of research. A straightforward approach—which is often applied, for example, in help systems for conventional option setting interfaces (see Gabrielli and Jameson [2009])—is to offer users explicit help with this problem, informing them about the consequences of particular choices or making recommendations for different types of situation. For example, it might be noted in the user's manual for the smartphone in the above example that text entry through speech doesn't work very well in some types of physical environment and with some types of text. This type of advice may sometimes run counter to the natural tendency to emphasize the positive aspects of a system and to be upbeat about the prospects for its use. On the other hand, when you consider that people who are encouraged to overuse the system may get unnecessarily frustrated and give up on it, it can seem worthwhile to give them some realistic advice.

Inadequate Control over Interaction Style

Users of interactive intelligent systems sometimes complain about not having enough control over the way in which the system interacts with them (figure 7). (Other reasons users can want to control an intelligent system include the wish to avoid privacy violations and the desire to ensure an adequate level of system performance; these types of control are discussed in the sections on privacy and imperfect system performance, respectively.)

Causes

Most mainstream interactive systems offer the user various ways of controlling the appearance and behavior of the system. The user of a spreadsheet

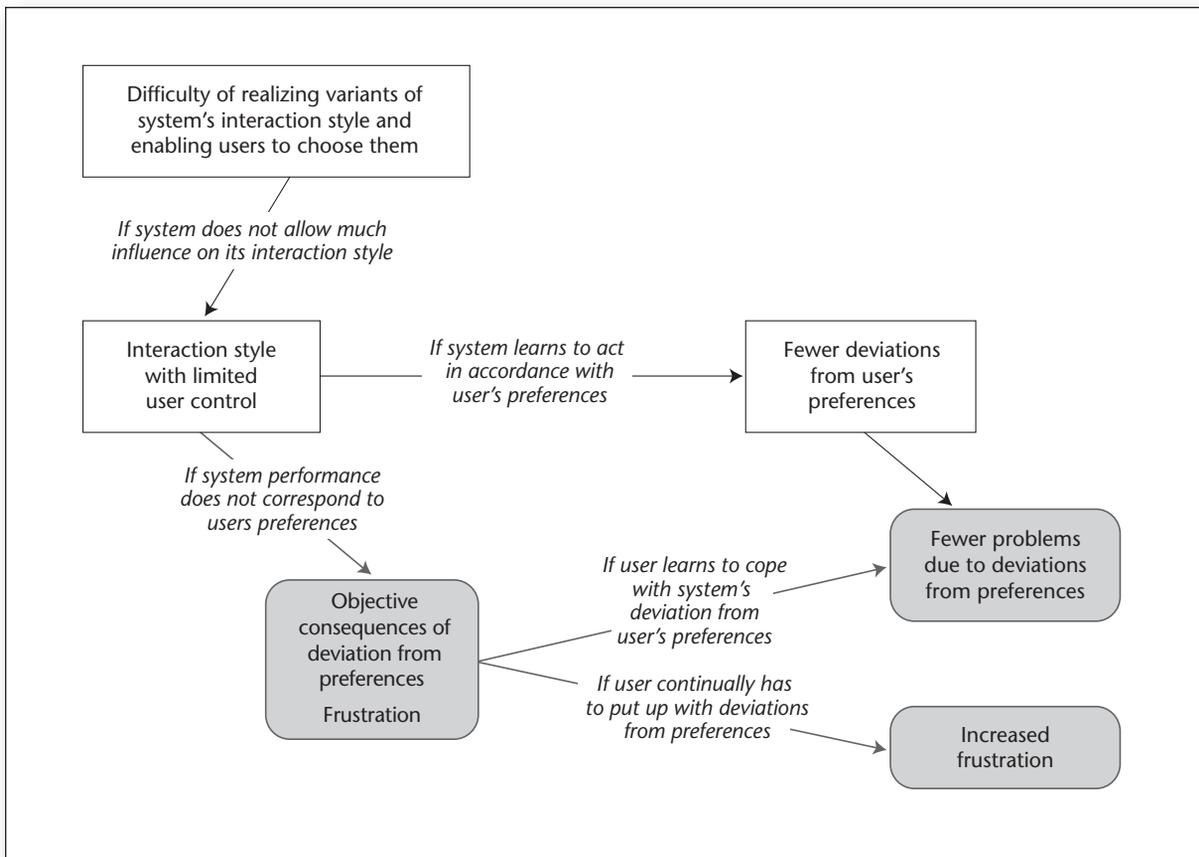


Figure 7. Analysis of Usability Side Effect in the Category Inadequate Control over Interaction Style.

can control features like the way in which certain types of information are formatted and whether the system will automatically recalculate formulas every time a change is made in the spreadsheet.

Where intelligent technology is used, customization is sometimes less straightforward.

First, since getting even one style of intelligent interaction right can be quite a challenge, it may be infeasible to offer the user much control over how the system interacts. For example, realizing two different dialogue styles in a spoken dialogue system may be much more difficult than offering only one carefully developed and tested dialogue style.

Second, it may be harder to devise ways of enabling the user to control the system's interaction style explicitly. For example, even if a spoken dialogue system has available several different strategies for confirming the user's input, there may be no convenient way for users to specify which one they want.

Consequences

When the system acts differently from the way the user would ideally like, the consequences can take different forms. The deviation may have specific objective consequences, such as a speed of presen-

tation that the user finds it hard to keep up with. Or the user may simply find the system's style of interaction somewhat irritating or distracting. A secondary consequence can be frustration at the fact that the aspects of the interaction in question cannot be changed, especially if it is not obvious to the user why it should be difficult to change them.

Changes over Time

In the most favorable case, the system may manage to adapt autonomously in a way that is consistent with the user's preferences (for example, adopting a more cautious interaction style if the user is having difficulty with the current style; see, for example, Litman and Pan [2002]). In this case, the negative consequences just mentioned ought to diminish over time. If the system continues to deviate from these preferences to the same degree, the adaptation on the part of the user may be either positive or negative: in the best case, users may get used to the aspects of the system's interaction style that they initially disapproved of. In the worst case, repeated exposure to a type of interaction that the user does not entirely like and is unable to change may lead to a buildup of frustration.

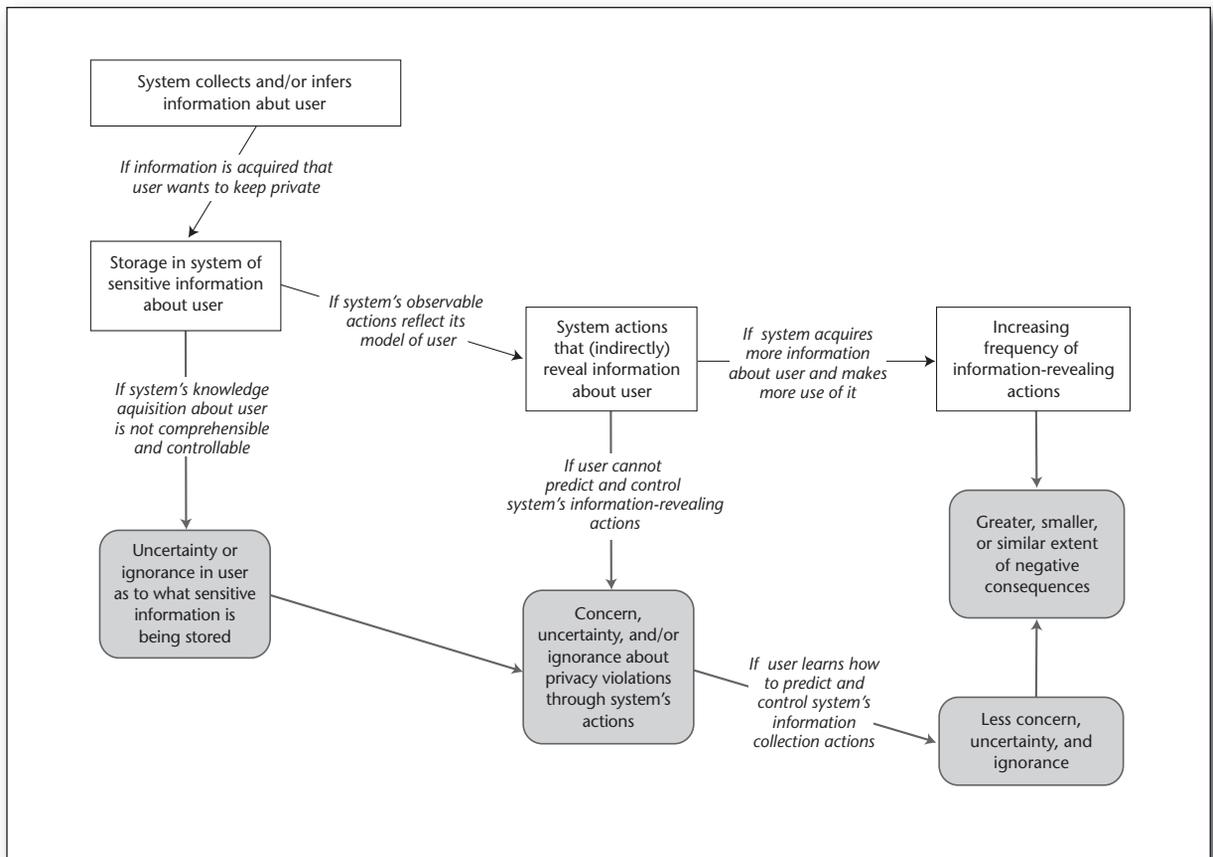


Figure 8. Analysis of Usability Side Effects in the Category Threats to Privacy.

Prevention

One approach to preventing a feeling of inadequate control over interaction style is to look carefully for the possible forms of control that do exist despite the obstacles mentioned at the beginning of this section. If there are at least some aspects of the system's interaction that can be controlled, users may be able to get at least fairly close to their preferred interaction style.

Second, it may be possible to combine autonomous system adaptation with simple user feedback (for example, "I don't like what just happened") to get around the difficulty of having the user specify the desired interaction style explicitly.

Finally, as a partial substitute for control, flexibility can be offered: the ability of the user to choose between alternative ways of doing something. For example, in the MICA system (described in the case study by Bunt, Conati, and McGrenere [2009]), which recommends user interface adaptations, the user can choose when to request recommendations, whether to accept all of them at once as opposed to deciding about each recommendation individually; and how much attention, if any, to pay to the explanations that the system offers of its recommendations. In this way, even though the

user cannot change any of the parameters that govern MICA's behavior, the user is able largely to determine the course of the interaction with the system.

Threats to Privacy

Many intelligent interactive systems gather and infer information about the user, sometimes with limited control and awareness on the part of the user, and sometimes including information that the user would refer to keep private. If the system's actions reflect in a visible way its user model, the user may justifiably be concerned that sensitive information will be revealed to others (figure 8).

Threats to privacy are one of the most frequently discussed negative side effects of both intelligent and nonintelligent interactive systems, because of their special practical importance. This section will not attempt to do justice to the large literature on this topic (see, for example, Cranor [2004] for a discussion of many generally relevant points within the domain of e-commerce personalization; and Kobsa [2007] for a comprehensive discussion of privacy and web personalization). Instead, we will see how this problem can be conceptualized with

our schema for the analysis of the side effects of intelligent processing, aiming to bring to the foreground points that have not been highlighted in most previous discussions.

The following analysis involves concepts that are discussed in connection with the overlapping classes of inadequate control over interaction style, inadequate predictability and comprehensibility, and need for learning by the user.

Causes

The intelligent functionality of an interactive system sometimes presupposes that the system is able to acquire information about individual users. For example, an intelligent agent that finds and presents news stories of likely interest to the user will have to know something about the user's interests, which may include topics like a medical condition that the user is suffering from. An adaptive user interface that modifies its appearance and behavior to accommodate a user's perceptual or motor impairments must have some information about these impairments. As these two examples indicate, some of this information may be of a type that the user prefers to keep private.

Consequences

The storage of sensitive information about a user is in itself by no means unique to interactive intelligent systems, but two additional typical features of intelligent processing can raise issues that are more specific to such systems:

First, the system may interpret behaviors of the user (for example, time spent reading a given news story or the timing of keystrokes during typing) without the user's knowledge of how they are being interpreted—or perhaps even of the fact that they are being interpreted. And it may make inferences on the basis of this type of input that it is hard for the user to predict or understand. As a consequence, the user may have at best a vague notion of the user model that the system has built up.

One likely concern here is that the information may become available to other parties in ways that the user would not approve of. Since security issues of this sort are well known and not specific to intelligent systems, we will not discuss them further.

A less obvious concern is that the system's observable behavior may indirectly reveal some of the contents of the model that the system has built up about the user. While a user is discussing a document on the computer screen with a colleague, for instance, an information agent might spontaneously display the title of a news story about a new medication. This problem is typical of intelligent systems, because many of them have an ability to adapt their behavior to their model of the user and to behave proactively.

This phenomenon may not be problematic if a system's privacy-violating actions are known to and controllable by users (for example, if users can turn off proactive notifications when a colleague is present, just as they might close the window of their e-mail client in that situation). But otherwise, users may justifiably be worried about the prospect of privacy violations if they continue to use the system; or they may remain ignorant of the danger until the first such violations have occurred.

Changes over Time

As with several of the other classes of side effect, there are two opposite temporal trends.

First, the amount of information that the system has acquired about the user is likely to increase over time, perhaps along with the system's ability to make use of it. Hence the opportunities for privacy violations are likely to increase.

Second, even if the possibilities for the user to understand and control the system's use of this type of information are limited, the user is likely to become more familiar with the potential privacy threats and to learn how to deal with them, thereby reducing the negative consequences.

The overall evolution over time will depend on, among other things, the relative strengths of these two tendencies.

Prevention

Several parts of figure 8 suggest high-level strategies for minimizing privacy violations.

First, limit acquisition and storage of sensitive information: a simple approach is to consider carefully, when designing the system, what sorts of potentially sensitive information really need to be acquired and stored, and for how long. Maybe the system can get by quite adequately if it purges all personal information once it is more than two weeks old—a policy that could make it easier for users to keep an eye on potential privacy violations.

Second, increase comprehensibility and controllability of the system's information acquisition. There are a variety of ways of putting the user into the loop with regard to information acquisition, though each of them has its limitations and drawbacks: (1) Acquire information through explicit queries, instead of through observation and inference. (2) Give the users at some point a general explanation of the information acquisition process. (3) Allow the user to check and edit the model that has been built up. The last two methods have been used, for example, in connection with Google's history-based personalized search.

Third, increase predictability and controllability of the system's information-revealing actions. Measures to consider here include: (1) Avoid unexpected proactive actions that might reveal sensi-

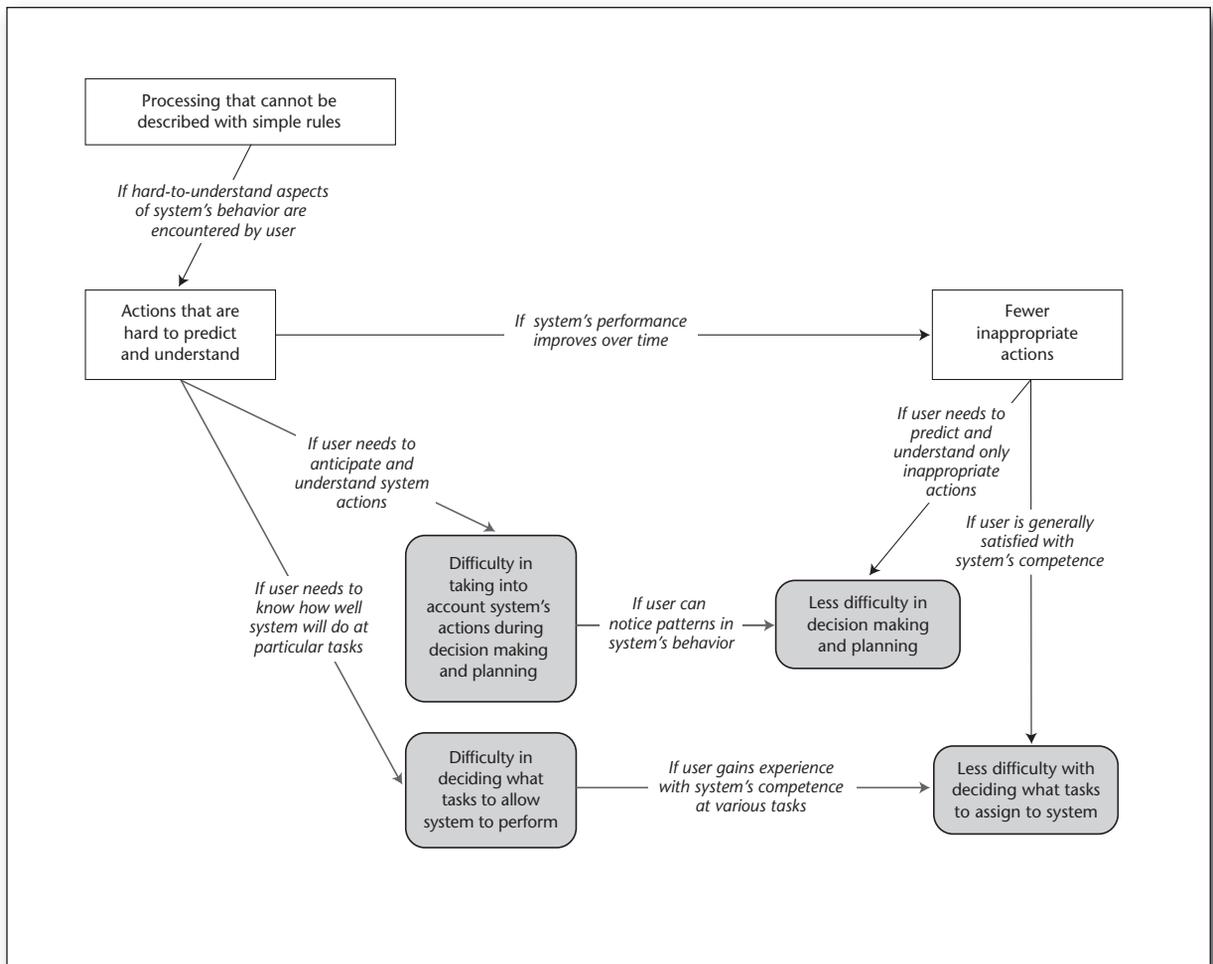


Figure 9. Analysis of Usability Side Effects in the Category Inadequate Predictability and Comprehensibility.

tive information (for example, instead of spontaneously displaying the title of a recommended news article, the system might just show a light bulb indicating that an interesting news article has been found). (2) Prefer presentation modalities that minimize observability by others (for example, small print on a screen as opposed to speech output). Since any measure of this sort may make the system less effective for the user, it may be worthwhile to introduce separate modes for situations where privacy is a concern versus those where there is no danger of privacy violations (for example, because no one else is nearby).

Inadequate Predictability and Comprehensibility

Users may be less able than with mainstream systems to (1) predict or understand the system's actions in a particular situation and (2) assess the system's ability to perform particular tasks (figure 9).

Causes

As is also discussed in the section on imperfect system performance, the processing methods and knowledge used by intelligent systems tend to make more complex the relationship between the input that the system receives and the actions that it generates. A user who types the letters "site" into a conventional word processor can predict with great confidence what letters will appear on the screen (unless the word processor includes an intelligent autocorrection feature). But if a user says the same word to a dictation system, it is hard to be sure which of the three words *site*, *cite*, and *sight* will appear on the screen, even taking into account the surrounding context. And if the word that appears is a different one than the user desired and expected, the user is likely to have a hard time seeing why this particular result occurred.

Many other types of intelligent system likewise operate with combinations of models, data, and algorithms that make prediction and understanding by users difficult. And the behavior of some

intelligent systems depends on variable environmental factors such as ambient noise and lighting.

Even if a user cannot predict and explain individual system actions, they may at least be able to assess the system's level of competence at a particular task (for example, the overall frequency of recognition errors of a speech recognizer). But even this type of assessment can be difficult, because the system's performance can vary from one situation to the next as a function of the particular content being dealt with or the particular situation in which the system is operating.

Consequences

Limited predictability and comprehensibility by no means always have negative consequences. For example, when users submit a query to an (intelligent) web search engine, they may not be able to predict what particular search results will be returned, or even the general nature of the results. An unexpected but relevant result may be accepted just as a pleasant surprise, even if the user has no idea how the system generated it. And an unexpected irrelevant result will not be considered worth trying to understand.

But the lack of predictability and understandability can be a problem if the user wants to plan ahead, taking into account the system's future actions. If the order of the items in a menu is constantly being adapted (see Findlater and Gajos [2009]), even an experienced user will not know in advance where to look for the desired item in the menu. When users of a sketching program are composing a drawing, they will in general want to know in advance what the system is going to produce in response to a particular drawing gesture; and if the system produces something unexpected, users would like to have some idea as to why, in order to be able to avoid a repetition of the problem.

On a different level, limitations in the user's ability to assess the system's competence can make it difficult for users to decide whether to use the system for a particular task.

Changes over Time

The types of understanding just discussed are often subject to considerable improvement as the user gains experience with the system. Even in the absence of any understanding of the basic workings of the system, observation of how the system acts in different situations can make it easier over time for the user to predict particular results and even to view exceptional results as being part of some pattern. Similarly, noticing how successful the system is at various tasks in various circumstances can provide a foundation for increasingly accurate assessments of the system's ability to perform particular tasks, even if the user has no idea

of how to explain the observed differences in performance.

Another facilitating factor is the tendency of many systems to improve their performance over time because of learning. Better performance can be easier to predict and understand than poor performance, which can involve strange deviations from the user's expectations; and there may be less need to understand and predict the details of good system performance.

Prevention

The first question that should be asked in connection with prevention is: What negative consequences—if any—do deficits in predictability and comprehensibility have with this particular system? The answer to this question can serve as a guide to the selection of preventive measures.

For example, with web search engines it appears that a lack of predictability of the search results is mainly a problem in cases where a user enters a search query with the intention of revisiting a previously visited page. In this case, it is helpful if the desired page consistently appears in the same position in the search result list. Accordingly, it makes sense to focus efforts to enhance predictability and comprehensibility on this case (see, for example, Teevan et al. [2007]).

Once it is known where these factors are important, various high-level strategies for achieving them can be applied.

One strategy is to consider choosing methods with good inherent predictability and comprehensibility. In the realization of intelligent functionality, there is often a choice between methods that make the system relatively easy to predict and understand and methods for which it is inherently more difficult. Some examples of differences of this sort are given in the article by Findlater and Gajos (2009), in this issue, as part of a systematic analysis of the design space for graphical adaptive user interfaces.

A second strategy is to enable the system to explain its actions. An increasing number of interactive intelligent systems offer some sort of explanation of the system's actions. For example, the commercial music discovery service Pandora (www.pandora.com), when asked why it is playing a given song, may offer an explanation like "... because it features a leisurely tempo, a sparse piano solo, a lazy swing groove, major tonality, and many other similarities identified in the Music Genome Project." Another example of an explanation facility is given in the case study by Bunt, Conati, and McGrenere (2009), in this issue. An explanation may make it clear why a given action of the system occurred, how the system works more generally, and how well it is currently performing; and this understanding can in turn

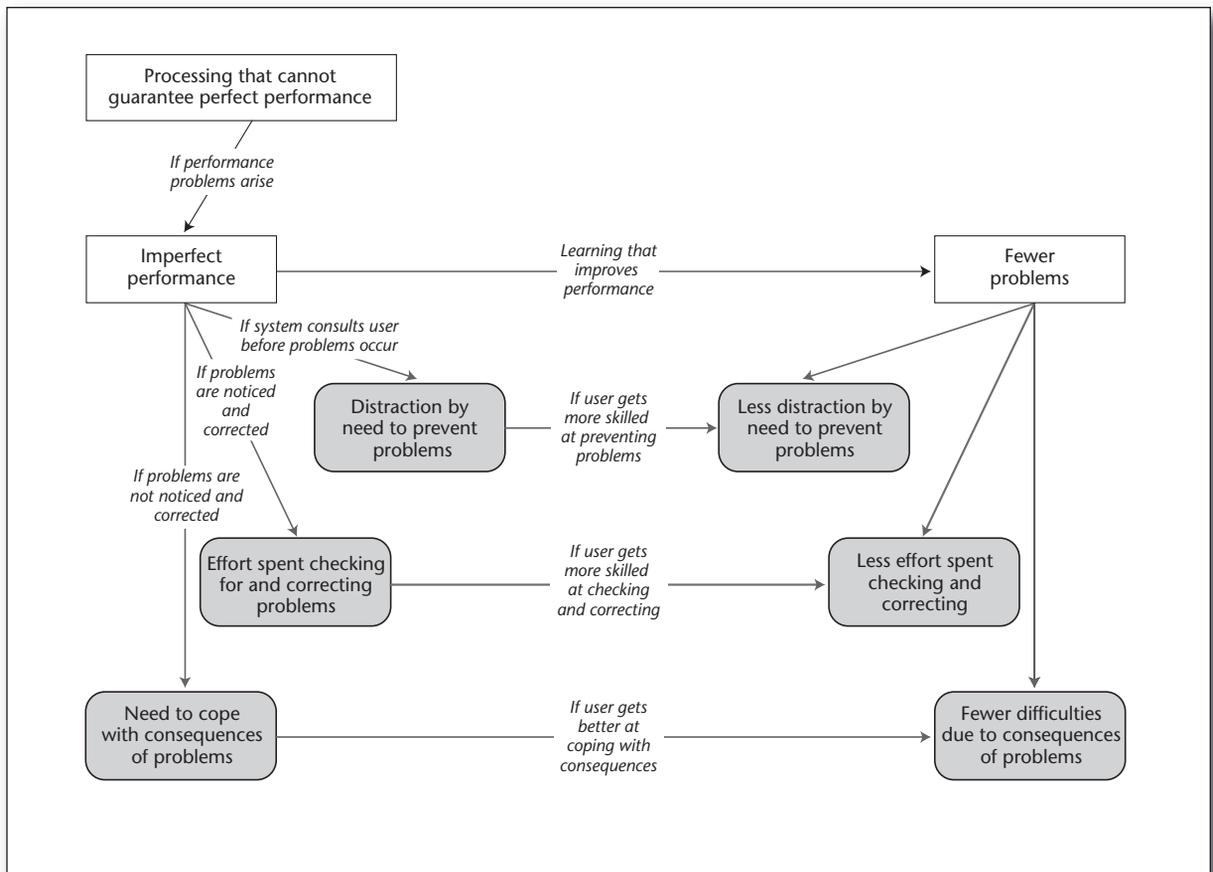


Figure 10. Analysis of Usability Side Effects in the Category Imperfect System Performance.

enhance predictability. Depending on the nature of the intelligent algorithms that are involved, there may be many possible forms that explanations can take, so it is important to consider exactly what the explanation is supposed to achieve. (See Tintarev and Masthoff [2010] for a more comprehensive discussion of explanations and the various functions that they can have, not all of which involve predictability and comprehensibility.)

Imperfect System Performance

A side effect of the intelligence of a system can be errors or other types of suboptimal performance that require some sort of correction or compensation by the user (figure 10).

Causes

Paradoxically, a frequent consequence of a system’s attempt to be intelligent is behavior that users are likely to see as stupid. The sort of processing that is typical of intelligent systems does not always guarantee accurate interpretation of all inputs or appropriateness of all system actions.

Consequences

The consequences of suboptimal system perform-

ance can take many forms, depending in part on the extent to which the problem is noticed by the user. For concreteness, consider a spoken dialogue system for travel reservation that mistakenly interprets the user’s request for a plane ticket from Los Angeles to Oakland, California, as a request for a ticket to Auckland, New Zealand.

In the best case, the system will have the user check its interpretation before it makes any use of it, so the negative consequence will be just the effort required by the user to correct the interpretation.

In a less favorable case, the user may notice the problematic system action only after it has been performed. In this case, the consequences can be more or less serious, depending on how difficult it is to compensate for the inappropriate action at that point.

If the user does not notice the problem in time to take any corrective action, there is in principle no upper bound on the seriousness of the consequences.

Imperfect system performance can also have a more subjective type of consequence: creating confusion in the user and reducing the user’s trust in the system. Since these consequences have been discussed in the section on predictability and com-

prehensibility side effects, we will not consider them further here.

Changes over Time

We can see two parallel tendencies, involving adaptation on the part of the system and the user, respectively.

First there is a general tendency for intelligent systems' performance to improve over time, as the system accumulates relevant knowledge. In this case, the negative consequences for the user will tend to taper off regardless of the user's responses.

Second, on the basis of experience with a particular system, users often become more skilled at anticipating, recognizing, and compensating for instances of suboptimal performance. Therefore, the consequences may become less severe even if the system's performance does not improve.

Both of these tendencies imply that suboptimal system performance is one of the problems in the early phase of use that may discourage users from reaching the stage of later use.

Prevention

The most obvious approach to preventing problems due to errors and other suboptimal behavior is to improve the performance of the system so that these phenomena are minimized. Indeed, a great deal of research on interactive intelligent systems takes the form of finding algorithms that perform better than previous algorithms according to some accuracy metric or other metric of performance quality (for example, quality of machine translation). Although this type of research is doubtless important, some additional considerations need to be borne in mind:

The consequences that various sorts of errors will have during the use of a particular system by particular users in a particular context need to be taken into account. The RADAR case study (Faulring et al. 2009) explains why, for beginning users, it was considered important to avoid false positives in the identification of tasks from e-mail messages, because of the confusion that they would be likely to cause. On the other hand, false negatives in this context would be likely to go entirely unnoticed, which could have drawbacks of a different sort. Sometimes, an intelligent algorithm includes a parameter that can be set differently in different contexts so as to ensure that the errors that occur tend to be of the least damaging types for that context (for example, a threshold of confidence that needs to be exceeded before the system will claim to have identified a task in an e-mail message). In other cases, new research may be required to find an algorithm that generates an appropriate distribution of error types.

As can be seen in the overview of the consequences of suboptimal performance in figure 10,

even if performance problems cannot be avoided, it may be possible to mitigate their consequences through appropriate interface design. The agent-assisted form-filling interface of RADAR is an example of careful interface design that aims to make it as easy as possible for users to detect potential errors before they have any consequences. As this example shows, the interaction design considerations involved in this approach may be quite specific to a particular type of interface and task. In the area of voice user interface design (see, for example, Cohen, Giangola, and Balogh [2004]), a great deal of experience has been accumulated with different approaches to the use of speech to request confirmation and to correct errors once they have occurred.

A more generic strategy is for the system somehow to indicate its confidence that what it is doing or suggesting is correct, as a way of helping users to detect and deal with possibly suboptimal system actions. But as is discussed by Kristensson (2009), this strategy does not always lead to improved detection of errors.

Because of the differences between early and later use mentioned previously, it can make sense to apply a different set of preventive measures during early use.

Concluding Remarks

The categories of usability side effects discussed in this article are not claimed to be exhaustive, and the specific analysis proposed of each class is subject to discussion and improvement. But the article should have shown that a full awareness of the usability side effects of intelligent processing does not need to result in gloom about the prospects of putting AI techniques more widely in service of users of interactive systems. On the contrary, the analysis shows how many different approaches are available to avoiding and mitigating the usability side effects that have to date so often impeded the acceptance of interactive intelligent systems. Some of these approaches are well illustrated in the case studies in this special issue; the cross-references in the case studies point to many examples that have not been mentioned in the text of this article.

Acknowledgments

The research described in this article was supported by funding from the targeted research unit Pre-revolution (code PsychMM), funded by the Autonomous Province of Trento, while the author was visiting FBK-irst. Judith Masthoff supplied valuable feedback on an earlier version.

Note

1. Also see the unpublished 1997 manuscript by A. Wexelblat and P. Maes, "Issues for Software Agent UI," available from citeseer.ist.psu.edu/128549.html.

Visit AAI on Facebook and LinkedIn

AAAI recently launched a presence on both Facebook and LinkedIn. We invite all interested individuals to check out the Facebook site by searching for AAI. If you are a current member of AAI, you can also join us on LinkedIn. We welcome your feedback on these new initiatives at info10@aaai.org.

References

- Bunt, A.; Conati, C.; and McGrenere, J. 2009. Mixed-Initiative Interface Personalization as a Case Study in Usable AI. *AI Magazine* 30(4).
- Cohen, M. H.; Giangola, J. P.; and Balogh, J. 2004. *Voice User Interface Design*. New York: Addison-Wesley.
- Cranor, L. F. 2004. I Didn't Buy It for Myself: Privacy and Ecommerce Personalization. In *Designing Personalized User Experiences in eCommerce*, ed. C. Karat, J. Blom, and J. Karat, 57–74. Dordrecht, Netherlands: Kluwer.
- Faulring, A.; Mohnkern, K.; Steinfeld, A.; and Myers, B. A. 2009. The Design and Evaluation of User Interfaces for the RADAR Learning Personal Assistant. *AI Magazine* 30(4).
- Findlater, L., and Gajos, K. Z. 2009. Design Space and Evaluation Challenges of Adaptive Graphical User Interfaces. *AI Magazine* 30(4).
- Gabrielli, S., and Jameson, A. 2009. Obstacles to Option Setting: Initial Results with a Heuristic Walkthrough Method. In *Proceedings of the Twelfth IFIP Conference on Human-Computer Interaction (INTERACT 2009)*, 400–403. Berlin: Springer.
- Gajos, K. Z.; Wobbrock, J. O.; and Weld, D. S. 2008. Improving the Performance of Motor-Impaired Users with Automatically Generated, Ability-Based Interfaces. In *Human Factors in Computing Systems: CHI 2008 Conference Proceedings*, ed. M. Burnett, M. F. Costabile, T. Catarci, B. de Ruyter, D. Tan, M. Czerwinski, and A. Lund, 1257–1266. New York: Association for Computing Machinery.
- Grudin, J. 2009. AI and HCI: Two Fields Divided by a Common Focus. *AI Magazine* 30(4).
- 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 *Human Factors in Computing Systems: CHI 1999 Conference Proceedings*, ed. M. G. Williams, M. W. Altom, K. Ehrlich, and W. Newman, 159–166. New York: Association for Computing Machinery.
- Iqbal, S. T., and Bailey, B. P. 2006. Leveraging Characteristics of Task Structure to Predict the Cost of Interruption. In *Human Factors in Computing Systems: CHI 2006 Conference Proceedings*, ed. R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, 741–750. New York: Association for Computing Machinery.
- Jameson, A. 2008. Adaptive Interfaces and Agents. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, 2nd ed., ed. A. Sears and J. A. Jacko, 433–458. Boca Raton, FL: CRC Press.
- Kobsa, A. 2007. Privacy-Enhanced Web Personalization. In *The Adaptive Web: Methods and Strategies of Web Personalization*, ed. P. Brusilovsky, A. Kobsa, and W. Nejdl, 628–670. Berlin: Springer.
- Kristensson, P. 2009. Five Challenges for Intelligent Text Entry Methods. *AI Magazine* 30(4).
- Lanier, J. 1995. Agents of Alienation. *Interactions* 2(3): 66–72.
- Lau, T. 2009. Why Programming by Demonstration Systems Fail: Lessons Learned for Usable AI. *AI Magazine* 30(4).
- Lieberman, H. 2009. User Interface Goals, AI Opportunities. *AI Magazine* 30(4).
- Litman, D. J., and Pan, S. 2002. Designing and Evaluating an Adaptive Spoken Dialogue System. *User Modeling and User-Adapted Interaction* 12(2–3): 111–137.
- Norman, D. A. 1994. How Might People Interact with Agents? *Communications of the ACM* 37(7): 68–71.
- Shneiderman, B., and Maes, P. 1997. Direct Manipulation Versus Interface Agents. *Interactions* 4(6): 42–61.
- Teevan, J.; Adar, E.; Jones, R.; and Potts, M. 2007. Information Re-Retrieval: Repeat Queries in Yahoo's Logs. Association for Computing Machinery 151–158. New York: Association for Computing Machinery.
- Tintarev, N., and Masthoff, J. 2010. Explanation of Recommendations. In *Recommender Systems Handbook*, ed. P. B. Kantor, F. Ricci, L. Rokach, and B. Shapira. Berlin: Springer. Forthcoming.
- Tsandilas, T., and Schraefel, M. 2004. Usable Adaptive Hypermedia. *New Review of Hypermedia and Multimedia* 10(1): 5–29.
- Weber, E. U., and Johnson, E. J. 2009. Mindful Judgment and Decision Making. *Annual Review of Psychology*, 60, 53–88. Palo Alto, CA: Annual Reviews, Inc.
- Ziegler, C.; McNee, S. M.; Konstan, J. A.; and Lausen, G. 2005. Improving Recommendation Lists through Topic Diversification. In *Proceedings of the Twelfth International World Wide Web Conference*, 22–32. Geneva: IW3C2.

Anthony Jameson is a principal researcher at DFKI, the German Research Center for Artificial Intelligence. He has been conducting research at the intersection of AI and HCI for 30 years, being among the first researchers to address topics like conversational recommender systems, Bayesian approaches to user modeling, adaptation to situational resource limitations, recommendation through decision-theoretic planning, and the temporal evolution of users' choices regarding interactive intelligent systems. A comparable amount of experience with application-oriented work on intelligent and nonintelligent interactive systems has convinced him of the central importance of dealing with the usability side effects of intelligent processing.