Steps to take before Intelligent User Interfaces become real

Kristina Höök
SICS
Box 1263, S-164 28 Kista, Sweden
+46-(0)8-752 15 17
kia@sics.se, http://www.sics.se/~kia/

ABSTRACT

Intelligent user interfaces have been proposed as a means to overcome some of the problems that direct-manipulation interfaces cannot handle, such as: information overflow problems; providing help on how to use complex systems; or real-time cognitive overload problems. Intelligent user interfaces are also being proposed as a means to make systems individualised or personalised, thereby increasing the systems flexibility and appeal.

Unfortunately, there are a number of problems not yet solved that prevent us from creating good intelligent user interface applications: there is a need for methods for how to develop them; there are demands on better usability principles for them; we need a better understanding of the possible ways the interface can utilise intelligence to improve the interaction; and finally, we need to design better tools that will enable an intelligent system to survive the life-cycle of a system (including updates of the database, system support, etc.). We define these problems further and start to outline their solutions.

Keywords

Intelligent user interfaces, user modelling, authoring tools, usability

1. INTRODUCTION

"Why should people have to adapt to systems, systems should adapt to people instead?" is a slogan that seems intuitively appealing to many users, as well as to researchers in the field of intelligent user interfaces. Unfortunately, the slogan may lead both users and researchers astray, making them believe that an intelligent interface should behave like a fellow human being, smoothly changing its behaviour to fit with users' knowledge, abilities and preferences, usually with advanced dialogue (and multimodal), capabilities. Contrary to that the very few intelligent user interfaces that have succeeded commercially have done either very simple adaptations based on simple knowledge of the user, or created its adaptations based on what *other* users do rather than some kind any complex inferred model of user. (The most well known examples of commercial success stories are perhaps FireFly[1] and Microsoft agent Lumiere [Horvitz, 1997] – they exemplify these two approaches.) Slowly, these new minimalistic intelligent systems are starting to show that they can help users to tackle information overflow/filtering problems, provide help on how to use complex systems, or take on tasks from users so that they can handle cognitive

[1] http://ww..firefly.com/

overload in real-time stressful situations. Intelligent user interfaces are also being proposed as a means to make systems individualised or personalised, thereby increasing the systems flexibility and appeal.

Even with these minimalistically intelligent interfaces, many human-computer interaction researchers are sceptical to the whole concept of intelligent user interfaces and personal assistants/agents, see [Shneiderman, 1997, Suchman, 1997]. Partly, the scepticism stems from experiences of previous artificial intelligence promises that have failed, but also from a fear that intelligence at the interface will violate usability principles and obscure the issue of responsibility. My claim is that the main problem is not whether or not intelligence at the interface is possible or desirable – this depends a lot on the task to be solved and the design of the total solution (with both adaptive and non-adaptive parts). Instead, I can see a number of problems not yet solved that prevent us from creating good applications. There is a need to develop:

- usability principles for intelligent interfaces (rather than direct-manipulation systems) that do not lead users' expectations astray
- reliable and cost-efficient intelligent user interface development methods
- a better understanding of how and when intelligence can substantially *improve* the interaction (design practice)
- authoring tools that enable easy development and maintenance of the intelligent parts of the system

Once we get a better grip on these problems, we shall be able to create commercially acceptable and useful (to the users) intelligent user interfaces that do not imitate human-human communication but instead aid the computer-human interaction process.

Below I discuss each of the problems in some more detail and I outline some potential solutions to these problems.

2. USABILITY PRINCIPLES

Recently the intelligent user interface community has started to worry about usability issues. This is exemplified by the discussions and panels at the latest intelligent user interface conference, [IUI'97], and also by the work on usable intelligent agents (or personal assistants). The attention has lead to the emergence of better problem definitions and some design criteria.

Control transparency and predictability

The problem with intelligence at the user interface is that it may violate many of the good usability principles developed for direct-manipulation systems. Those principles include giving the user *control* over the system, making the system *predictable* so that it always gives the same response given the same input, and making the system *transparent* so that the user can understand something of its inner workings. Systems that adapt to their users and changes their behaviour to better fit users' needs will by necessity violate, at least, the principle of predictability and possibly also not be transparent and may hinder users' control over the system.

Within the field of user modelling and intelligent interfaces, some approaches to tackling these problems have been proposed.

Understanding how adaptivity works (transparency) does not necessarily mean that the system has to explain exactly what it is doing in all its details. In our work, the PUSH project, we have applied the metaphor of a "black box in a glass box" [Höök, 1996, Höök et al. 1996]. In the PUSH system we hide the complex inferencing of users' goals (in the black box) and show a quite simplified view on what is going on (in the glass box) to the user. The user sees a straightforward relation between inferred goal (as unobtrusively presented to the user) and choice of adaptation, providing a sense of predictability. What the user does not see is exactly how the goal is inferred from his/her actions at the interface – in our system this is a quite complex relation.

John Seely Brown, (1989), talks about three different *glass box levels* in the context of tutoring/learning systems:

> ”The goal of design of any tool or device, therefore, should be to produce ‘glass boxes’, which, first and foremost, connect users to the real world. Further, in response to examination and investigation, they should allow users to build adequate mental models and provide useful focus for collaborative discussions and the social construction of knowledge. Essentially, the current opaque technology or ”black boxes” must become ”transparent” to the user, allowing him or her to see ”through” the tool (”domain transparency”) or ”into” the tools (”internal transparency”), or to see the relationship of the technology and its users in the larger context of the interaction between the user and the tool (”embedding transparency”). ”

By *domain transparency* we understand tools that allow the user/learner to see through the tool and see the domain behind it. For example, a tool that helps an auto mechanic service the ignition should allow the mechanic to see the ignition system through the tool. Or even work as a magnifying glass bringing the working of the domain into coherent focus.

Internal transparency is concerned with the tool itself and how the user can see through the tool’s interface into its internal workings. So the auto mechanic would not only see through the tool into the ignition, but also be allowed to learn parts of how the diagnostic aid itself reasons. This is the same perspective put forth by du Boulay, O’Shea, and Monk, (1980), when they utilise the glass box metaphor for programming languages: the programmer must be allowed to understand the execution mechanism of a programming language at some level of abstraction, in order to become a good programmer.

Finally, the *embedding transparency* refers to the whole environment in which the tool is going to be used. As John Seely Brown puts it:

> ”Technology design must concern itself with ways to remain connected with the world so that the interactions with the technology take place within the context of on-going interactions between the user and the world”.

The same principle basically applies to any computer application, we never see the whole picture of what is going on within a system. For intelligent interface, the big issue becomes what to place in the glass box and what to place in the black box(es). The glass box may well be a metaphor used to convey the state inside the adaptive system, as done with, for example, Patti Maes’ cartoon drawings of an agent watching the user work, [Kozierok and Maes, 1993].

Giving users control over the adaptivity can be done other ways too. Cook and Kay, (1994), proposes that users should be allowed to inspect and alter the user model in the system. Depending upon the individual user’s experience of computer systems and, in particular, user models this may of course be more or less difficult. It is not always obvious what aspect of the adaptivity that is changed when ha parameter in the user model is changed. An example is the *adaptive prompts* system by Kühme et al. (1993). This design allows a user, or a program analyst, to tailor the mechanisms for adaptivity, but in order to do this, the user must learn a new, complex vocabulary which distinguishes between sets of terms such as e.g. ”goal”, ”action”, and ”interaction”. Using these concepts, the user is supposed to construct a set of rules rudimentary guided by an interface. Apart from having to understand the meaning of concepts as goal, action , etc., users will have to predict the *effects* of this tailoring, as the different parameters and rules interact in a complex way to achieve adaptiveness.

Allowing the user to change the user model also introduced another potential problem in that it diverts the users' attention from their primary task and forces them to build a (sometime complex) model of the system's adaptive behaviour. Still, this kind of solution is useful in certain circumstances, e.g. when the model holds the user's preferences and those are expressed in a language that makes sense to the user community. In general, we know that few users spend time adapting their systems, so expecting them to spend time adapting the adaptivity is often unrealistic.

Another way of providing control to the user is through making the adaptivity part of the total design already from the start and making sure that there are means for the user to directly or indirectly correct the system's choices for adaptation. If, for example, the system is filtering information, it may do well in making the rejected information pieces available (somehow) for the user to look at. They can, for example, be greyed, or placed further down on the priority list, and available to the user if s/he wants to access it. This model was applied in the PUSH project [Höök, 1996, Höök et al. 1996, Höök, 1997].

In addition, many intelligent user interface designs split the interface into one part which is predictable and one part which is adaptive and unpredictable. The adaptive part is not allowed to manipulate the non-adaptive parts of the system, but only suggest possible actions or provide relevant information. This approach is used by, for example, the Letizia system [Lieberman,1995] that filters web-information. Letizia will not interfere with users web browsing, but will present interesting links in another window. The same approach is used in Alexa[2], Patti Maes systems [Kozierok and Maes, 1993], MS Office Assistant [Horvitz, 1997], among others.

Privacy and Trust

Apart from finding (roundabout) ways to give users a sense of control, through making the system seemingly predictable or transparent, we also need to tackle a couple of other problems that become more acute in this field than in general interface/system design. One is the privacy issue. All systems containing a user model forces users to accept that the system holds a representation of (some aspect of) them. Some of the intelligent user interface systems require furthermore that users are willing to share their preferences with a user community. Depending upon how anonymous we allow the user to be, this can be more or less of a problem. Let me provide two examples to illustrate that it is sometimes a problem and sometimes not.

In the FireFly application users are required to share their preferences for music or movies. Based on those preferences, the system can recommend new music or movies. The system provides its recommendations based on clustering of the preferences provided by the users. Users are allowed to be anonymous and they willingly provide their preferences for music to this system. In the Doppelgänger system on the other hand, providing a personalised newspaper [Orwant, 1994], the situation is somewhat different. Here the user can configure their newspaper through a user model that they can inspect and alter. They can ask the system to provide them with the same kind of news that, for example, a colleague of theirs is reading. This system was not acceptable to users - they did not want to share their user model with each other. (That this is an issue that must be dealt with seriously can be seen in, for example, the WiseWire privacy policy[3].) The difference between these two systems lies in two aspects: one is the anonymity of the user and the other is in how touchy the subject area is – music preferences do not seem as important as what we decide to read or not read of the news.

---

[2] http://www.alexa.com

[3] http://ww7.firefly.com/

Judy Kay has proposed a means to overcome this problem by splitting the user model into two parts: one public and one private [Cook and Kay, 1994]. So in applications where single users may feel that their privacy is threatened or may be misused, they can keep certain facts about themselves private and others public. Depending on the application the division between private and public can either be preset or determined by the user interacting with the model. Again, in many situations, the user can't be bothered to make these choices and so the preset solution is to be preferred.

Trust in the system is yet another problem that needs to be tackled by intelligent user interface designers. This problem is most prominent in the situations where the system takes on tasks from the user. If the system starts sorting our mail, filtering our news, retrieving information for us on the web, selling and buying goods on our behalf, etc., it will be of crucial importance that we can trust the system [Maes, 1995]. These problems have very little to gain from traditional solutions to security such as encryption algorithms, etc. Within the multi-agent community solutions are proposed that rely on "social" protocols as a means to find out who can be trusted. For example, before my agent buys something from another agent, my agent can be allowed to check which other agents the selling agent has traded with before, and in turn, check if those agents can be trusted[4]. For agents that interact with the user, the same kinds of interactive, social, solutions can be envisioned: the user can be notified at certain crucial points when the agent adds knowledge or rules about the user or applies a rules for the first time, etc. So rather than relying on the system handling security, some responsibility will be laid on the user to react on "warnings" from the system.

Unfortunately, we know that if an adaptive system gives the wrong advice just once, users' trust in the system will go down drastically, and they may not use the system for a long time. Partly, this can be a question of culture. Once we get used to having adaptive systems around us, we will also gradually build models of how they work and when they can be trusted.

Treating Systems as Fellow Beings

There are also other, secondary, effects of using intelligence in systems. They may change the users' understanding of computers as stupid machines, and shift users' perception of responsibility from users to systems. Shneiderman [1997] and Lanier [1996] argue against antropomorphic agents since they give users the impression that the system will be able to take responsibility for its actions and that they will act rationally, similar to another human being.

In my view, it is of crucial importance that the design of the intelligent agent (or personal assistant) is such that it creates the right expectations in the user: neither too high nor too low. Here we must be careful of imitating human-human communication and assuming that that would be the best model of performance. Agents or adaptive systems should be implemented so that they become an integral part of an otherwise well-designed system. Furthermore, the design must be such that their interaction is most beneficial to the tasks that the system and the user are attempting to solve together.

Suchman [1987, 1997] proposes that the term "interaction" might best be reserved to describe what goes on between persons, rather than extended to encompass relations between people and machines. Her argument is based on the assumption that any conceptual model that a designer will build into an interactive system will have its limitations and in many cases, will not at all reflect what users actually use the system for. So any agent or adaptive system that makes any assumptions on what tasks the user will be

[4] A solution along these lines has been proposed by ??. His agents will collect signatures from other agents that are content with their interaction with them. By following the links from these signatures onwards to those agents, their signatures can be checked in turn.

attempting to solve and that tries to interact with the user from these assumptions will fail. Her proposal is instead to make machines that are "readable", i.e. where the user can read the system's state and interpret those himself/herself and then act.

Suchman's criticisms stems from her experiences with early artificial intelligence research at Xerox, and though her criticism may be quite valid for early expert systems design it does not seem as valid when we talk about, for example, recommender systems [Recommender Systems, CACM, 1997]. Recommender systems will not base recommendations to a particular user on some designer-constructed conceptual model with invented "rules", but instead on single user or whole user population's behaviour. It is in fact users' choices that will constitute the advice. The most interesting future for intelligent user interfaces lies in finding the best way of extracting people's preferences and intelligent behaviour and using this as the basis for adaptation. It does not have to be based only on large user population behaviour (as in [Lashkari et al. 1994]), but can also draw on single, expert, user's behaviour (as in [Höök et al. 1997]). Most importantly, it must be constructed to be flexible and able to change over time as user behaviour changes.

Even if we can find examples of intelligent user interfaces that will fulfil the "readable"-criteria discussed by Suchman, a main point with many intelligent user interfaces and in particular, agent systems, is the fact that they reintroduce the dialogue between user and system. These intelligent user interfaces move away from direct-manipulation, and instead are manipulated indirectly. In some sense, it is a move back to the time when we had command-based interfaces, only this time the system response is supposed to be adapted to the user. In my view, the interactivity or dialogue capability of an intelligent user interface is not necessarily a bad thing if it can sensibly convey its limitations. A dialogue with a computer should not look or feel like a dialogue with a person. Any assumptions of users' tasks should be transparent to users.

I believe that the fear of alienating users through not providing them with insight into how the system works, is not a problem that is unique to intelligent user interface's. Most computer applications function like this - at least for those who are not computer scientists or particularly knowledgeable in how the system works. As pointed out by Patti Maes, (1997), the same argument could be made against driving cars when you do not have a complete model of how the engine or the brakes work. The point is that people do drive around in cars and manage really well without these complete models. When the car breaks down, they willingly hand over the car to a car mechanic agent whom they have to trust to have the necessary skills for fixing it. The point is that the main task for the driver is to use the car the get somewhere - they can't be bothered with a complete model of the car's internal workings. The same goes for computer systems - if they can provide me with, for example, high-quality relevant information, I would not necessarily be bothered by exactly how it is done (just as I do not (usually) bother myself with how the journalists of my daily newspaper find the information).

In summary, usability of intelligent user interface is an important and crucial problem to be addressed by the research community. The attempts to tackle the problem are promising but they require a new way of addressing usability, sometimes quite different from the usability principles outlined for direct-manipulation systems. To the list of problems, we also need to add privacy, trust, and responsibility, issues that have not been much addressed in the human-computer interaction research community before.

3. METHODS

When designing, in particular, adaptive help systems, it is quite common to discover that many problems could be overcome through redesigning the target system [Breuker, 1990]. In my view, this problem points at the necessity to make the adaptive (or intelligent) parts of the system into an integral part of a good total design of the system. The whole system design should meet users' needs - the adaptivity should not fix a bad design. As phrased by Eric Horvitz (in a panel at Intelligent User Interface'97 [1997]):

"We must remain alert to attempts to use sophisticated inference simply to get around poor design, or in lieu of better design combined with simple automation techniques."

This means that the issue of whether or not some part of a system should be adaptive or intelligent must be part of the design process already from the start. This is rarely the case in system design as it is done today – and it will not be until we provide efficient methods for how to integrate these ideas into the design process.

Also, as pointed out by, among others, David Kurlander, it is critical to weigh the advantages of using artificial intelligence techniques versus the benefits of doing things traditionally. Intelligent interfaces often make mistakes and they may be slow. We must be able to say when and how intelligent solutions fit into designs.

As long as intelligent user interface's are developed by highly skilled academics the need for methods is not crucial. But as soon as we want to move the design of these systems out into industry the availability of methods, tools and standards will be the crucial test that determines whether they will succeed or not.

Unfortunately, methodology for domain analysis from the adaptive systems' perspective remains largely a field in its infancy [Benyon, 1993]. Researchers in adaptive systems often make claims about user needs that have very little to do with what will actually be of real help to users. A proper analysis of users, their tasks and needs, is therefore a necessary part of any development of an adaptive system.

Benyon discusses five analysis phases that need to be considered when designing adaptive systems:

- *functional analysis* aims to establish the main functions of the system.
- *data analysis* is concerned with understanding and representing the meaning and structure of data in the application. Data analysis and functional capabilities go hand in hand to describe the information processing capabilities.
- *task knowledge analysis* focuses on the cognitive characteristics required of users by the system, e.g. the search strategy required, cognitive loading, the assumed mental model, etc. This analysis is device dependent and hence requires some design to have been completed before it can be undertaken.
- *user analysis* determines the scope of the user population that the system is to respond to. It is concerned with obtaining attributes of users that are relevant to the application such as the required intellectual capability, cognitive processing ability, and prerequisite knowledge required. The anticipated user population will be analysed and categorised according to aspects of the application derived from task, functional, data and environment analysis.
- *environment analysis* covers the environment within which the system is to operate. This includes physical aspects of the environment and 'softer' features such as the amount and type of user support that is needed.

As pointed out by Benyon there are few attempts to providing methods for user- and environment analysis. The latter might potentially benefit from approaches such as situated cognition [Suchman, 1987] or activity theory [Nardi, 1996]. In those approaches, ethnographic or anthropological methods are used to study users in their normal working environment. Unfortunately, those are quite time-consuming (several months of study may be required), and as of yet they have not proven that the cost is worth the effort.

User analysis is of course crucial when we design systems that maintain a user model. A user analysis

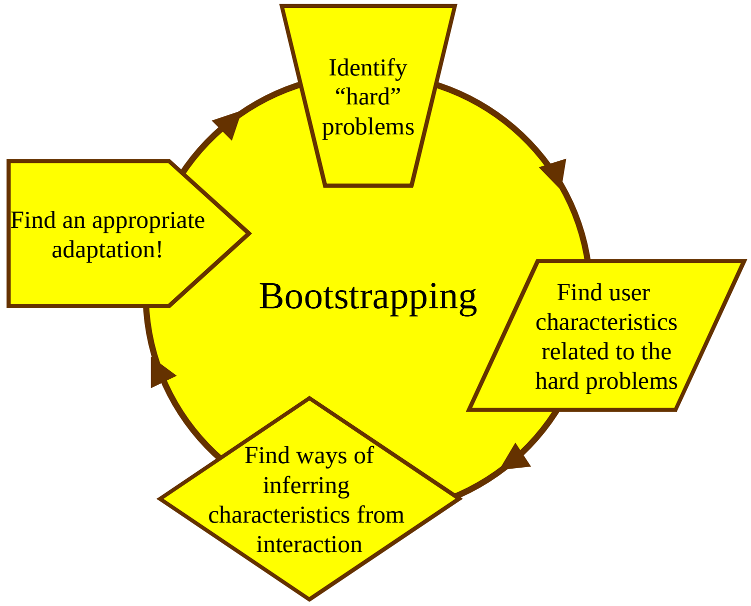

Figure 1. Bootstrapping the adaptive system.

should preferably arrive at a characterisation of the targeted user group that can be linked to the problems the designer hopes to solve through an adaptive behaviour. Furthermore, the relevant characteristics found in the user community must be inferable from their interaction with the system. That is, if the system shall utilise indirect inference of user characteristics from their interaction with the system. If, on the other hand, designers want to query users and directly get the sought characteristics, they must make sure that the queries will in fact render reliable results. (As pointed out already by Rich, (1979), users are not always a good source of information about themselves.)

What we get is a circular problem: we must identify a problem that would be solved by adaptivity, we must identify characteristics in the user group that would remedy the problem as well as be inferable from their actions at the system, and finally, we must identify an adaptation technique that will render the right kind of adaptive behaviour, see Figure 1. As pointed out by Opperman, [1994], this is a design process that requires a bootstrapping method: first some initial design of the adaptive behaviour is implemented which is then tested with users, revised, tested again, etc. The reason is that it is very hard to foresee how users' actions should be linked to particular adaptations.

Furthermore, the adaptations must be found to be of real use. Parts of this bootstrapping procedure may be automated using some machine learning technique, but not all of it since is the alternations you may want to do to the design might well be much more fundamental than just adjustments between user actions and corresponding adaptation. You may want to change which adaptations should be possible, which user actions should be monitored, etc.

A major problem with task analysis is called the "paradox of change" [Downs et al. 1988]:

"Current practice in a task analysis is frequently tied to the existing technology employed in the task and it is therefore difficult to produce a creative, novel solution to system design based on such methods."

So, in order to apply task analysis, there must be a tool that is used by a user population that we can study. If there is, our solutions might be too influenced by the existing organisation of work, tools used, etc.

In our experience, the best approach is to combine the user and task analysis into one step. In the user analysis, it is important that the designer has an open mind in searching for the relevant user characteristics. Preferably, aspects of both users' knowledge, users' goals (or tasks), users' preferences, and users' cognitive abilities and personality should be analysed. These must then be linked to the problem isolated that is in need for an adaptive solution.

Cognitive Task Analysis

A source of inspiration in the development of methods for intelligent user interfaces is the Cognitive Task Analysis (CTA) method developed by Roth and Woods, (1989). CTA was developed as a reaction against the *iterative refinement process* that was the most prominent method for development of knowledge-based systems in the eighties. The iterative refinement process starts out from a few example problems as defined by some expert in the field. It analyses how the expert solves those, quickly attempts to implement a prototype tested with new cases and shown to the experts. This provokes another round of refinement of the prototype to cover more example problems. The goal is to end up with a system that covers all the possible problems.

Apart from the obvious problem that the iterative process will take quite some time, Roth and Woods point to the danger of it causing the designer of the knowledge-based system to make faulty decisions too early on in the development. If the identification of the most prominent problems to solve, the system design, the knowledge representation and other design decision are made based on a few examples given by the expert initially, the solution might not scale up to the whole problem scenario.

Roth and Woods also point at the fact that the expert's problem solving behaviour might not be optimal. It might be the case that there is a lack of information, which causes the expert to behave in a certain manner. Given a better basis for making decisions, the expert might perform much better. So, in some cases, the underlying system / environment / information source must be changed first, before it is possible to find the optimal problem solving process in the domain.

Going back to the figure shown above, Figure 1, we can see that intelligent user interface development runs the same risk as knowledge-based systems did. It is easy to imagine how we develop some adaptive behaviour based on early assumptions on what will be of use to our users, perhaps based on some initial interviews with users. The early decisions made at this point may then misguide the whole intelligent user interface design process.

For example, rather than implementing the Microsoft Office Assistant [Horvitz, 1997] as is, a proper analysis of the environment, tasks and users of MS Office might have revealed that a whole other design with more integrated help facilities and scaled down functionality, is needed.

Don't Diagnose What You Cannot Treat

A proper analysis of users and their characteristics will reveal many individual differences that we should not necessarily use as the basis for adaptive behaviour in the system.

According to Karen Sparck-Jones, (1991), modelling the user can be done in a strong sense where characteristics not necessarily relevant to the functional task for which the system is designed are modelled, and modelling in a more restricted sense limited to those characteristics that are relevant to the system's task. What I would like to propose with the model outlined in Figure 1 is even more restricted since I only want to model such characteristics that can help us solve really hard problems where non-adaptive solutions fail.

Our approach here follows what John Self has put forth in the area of student modelling in intelligent tutoring systems (1988). Self expresses his critique as ”don't diagnose what you cannot treat'”. His critique comes from the fact that so much research effort was put into trying to infer rich and complex models of learners' understanding of some domain, and not enough effort was spent on figuring out what to do in order to tutor learners based on the diagnosed problems. We can paraphrase this in the area of user modelling as ”don't model user characteristics that do not (profoundly) affect interaction”, or perhaps even ”only model such characteristics of the user which cannot be catered for by other means”. If we can find ways by which users can control and alter e.g. a provided explanation so that it fits with their knowledge just by making the interface interactive and flexible, that is probably better than making the system guess at users' knowledge or other characteristics and be adaptive.

Unreliable User Characteristics

I proposed above that when performing the user analysis the designer should look for important individual differences linked to the problem to be solved. I divided those user characteristics into users' knowledge, users' goals (or tasks), users' preferences, and users' cognitive abilities and personality. These categories are quite different in terms of stability over time, how easily they can be inferred from the interaction with a user or user group, the reliability of the inferred knowledge, etc.

When modelling users' knowledge the reliability of the information inferred is a crucial factor. A user's knowledge is not static – it keeps changing, we learn, we forget, and we make mistakes for other reasons than lack of knowledge (like getting tired or being distracted by other tasks). Most models of users' knowledge will be unreliable [Kay, 1994]. The methods available for extracting a particular users' knowledge have two main deficiencies.

First, as pointed out by Yvonne Wærn and colleges, (1990), is that the interaction with users has to be very rich (using natural language and other interaction means), if we are going to be able to infer anything from their interactions. Given limited communication channels, very few inferences can be made. Annika Wærn (1996) also points out that the actions of users are frequently at a very low level compared to what the system wants to infer about them.

Second, a common underlying assumption behind the classification of users' knowledge is that users tend to learn domain knowledge in a predictable order. Unfortunately, this is not true for all domains. For some domains, most of the information would be learnt only if needed. For others, knowledge may very well be learnt in several different ways. Furthermore, most classifications of domain knowledge is only based on classifying concepts, not on more complex aspects of knowledge, such as how different concepts interact or can be applied. Finding the order by which knowledge is acquired by a “normal” user is very difficult and time consuming. That in itself is a clear signal that this route of modelling users' knowledge will be difficult to follow, in particular in industrial settings where users are busy and cannot be bothered with extensive studies during the design of the adaptive system[5]. This is probably part of the explanations as to why we do not see many such systems in use.

Stereotypical models of users' expertise seem to be providing more mileage to system builders, but those are on the other hand blunt tools that often make faulty assumptions.

Unfortunately, the same unreliability as with models of users' knowledge is true for models of users'

[5] I am here disregarding the whole field of intelligent tutoring systems (see e.g. Wenger, 1987), where it has indeed been proven that it is possible to design tutoring systems that will hold a fairly reliable model of learners' knowledge. But those systems are built for very limited domains, and are usually designed to be used within educational systems rather than as aids in other contexts where users have more diverse backgrounds. Furthermore, the same problem concerning the time it takes to study users and find out about their development of knowledge holds for intelligent tutoring systems.

goals and plans. The emerging theories of situated cognition, constructive cognition, and distributed cognition, challenge both the goal-plan-oriented view on human behaviour and the previously held symbolic view on cognition [Suchman, 1987]. Users may not be as goal-oriented and rational, as some of the adaptive features of proposed systems require. People will act based on the situation they are in right now, so their goals and plans keep changing in response to how the situation changes and develops. If the adaptive system assumes a too rigid and static model of the user's plans and goals, it will not be able to capture the ”continuous improvisation” that people are involved in [Suchman, 1987]. There are some promising attempts to create less static models, and also to make the plan recognition more integrated with direct-manipulation interfaces, e.g. [Waern, 1996].

Inferring users' personality or cognitive abilities is a potentially fruitful, but, so far, little explored area. Some of these characteristics are quite stable, such as users spatial ability [Benyon and Murray, 1993] while others are less stable. Furthermore, these characteristics are quite hard to infer from users' interactions with the system.

Finally, modelling and adapting to users' preferences seems to be the most successful approach. Even if preferences are just as hard to infer from users' interactions with system, it is possible to ask users' about them and get reliable answers. Still many questions remain unsolved when it comes to distinguishing between long-term interests (such as those a researcher may have in a particular research field) and short-term interests (such as searching for information on where to go on vacation this year).

So, the problem of identifying user characteristics that are related to the problems that the adaptiveness of the system aims to solve, must also consider that some of the characteristics of people are neither stable nor easy to infer from users' interactions with the system. Again, if the adaptive parts of the system are designed in parallel with the design of the whole system, we stand a much better chance to succeed. We can then make sure that the allowed users' interactions with the system are such that the adaptive system can (without bothering users too much) infer what is needed.

4. DESIGN PRACTICE

One of the most important challenges for intelligent user interfaces is to prove that their adaptive behaviour does in fact *improve* the interaction with the user. Only through designing useful adaptations and then evaluating them with users can we be sure that we are solving the right problem. If we had many such studies showing that certain adaptations work, we could start to extract general principles for design. But evaluating systems is a difficult task, and it becomes even more difficult when the system is adaptive, and very few such studies have been performed so far.

When studying an adaptive system it is of crucial importance to be able to distinguish the adaptive features of the system from the general usability of the designed tool. This is probably why most studies of adaptive systems are comparisons of the system with and without adaptivity [Boyle and Encarnacion, 1994, Brusilovsky and Pesin, 1994, Kaplan et al. 1992, Meyer 1994, Höök 1997]. The problem with those studies is obvious: the non-adaptive system may not have been designed ’optimally’ for the task. At least this should be the case since adaptivity should preferably be an inherent and natural part of a system – when taken out the system is not complete. Still, it is very hard to prove that it is actually the adaptivity that makes the system better unless that condition can be compared with one without adaptivity.

In fact, several of the few studies made fail to prove that the main activity that the systems are supposed to support is improved by the adaptive behaviour. Instead these studies measure other, perhaps not as crucial factors as task completion time, number of errors, or (in hypermedia) number of revisited nodes. Just as with direct-manipulation interfaces, it is not always the task completion time that is most important to measure [Gilmore, 1995]. Or, in the case of educational hypermedia, it might very well be beneficial to the learner to keep revisiting concept explanations from different contexts as they thereby repeat the information and learn it. So measuring the number of revisited nodes and claiming that the fewer revisited nodes the better might very well be misleading. If the adaptive system is aimed at an educational purpose, then learning and the learning process is what should be measured. Or if the system should do information filtering, then we must check whether subjects find the most relevant information with the adaptive system and not necessarily whether they find it fast. This is not to say that the traditional measurements are always wrong – this of course depends upon the task that user and (adaptive) system should solve together.

But even when if we would succeed in creating many good adaptive applications that are properly evaluated, this would not necessarily give us the best basis for extracting good design practice. Most intelligent user interfaces, such as e.g. user agents, are designed with the aim of being of use in the long run – not only during a short, controlled, user study. Proper evaluations of whether the system supports users' real tasks must include an analysis of the organisational setting, users' activities and cooperation with each other, usage of other tools, etc. Activity theory might give us some of the tools needed to analyse this complex situation [Nardi, 1996, Cole, 1996].

So, the real test for intelligent user interfaces is whether they continue to be used after the initial excitement is gone.

5. SCALE UP!

The problem of scalability of artificial intelligence systems has been known for a long time [Schank, 1991]. Artificial intelligence systems have typically been developed by academics who make them work for a couple of examples, and then those systems cannot be scaled up to the cover the whole problem. The reason lies partly in the lack of tools for adding more knowledge to the system, but also in that artificial intelligence systems have not always tackled the right problems. As such systems are put to use, they turn out not to solve the real problems in the domain.

The new trend, known as new artificial intelligence, with intelligent agents and using machine learning and other techniques that moves us away from the issues of knowledge representation, inferencing, etc. have come to represent a possible way out of the scalability problem. These systems can supposedly learn and extend their coverage by themselves. But, in fact, some of the scalability problem remains as discussed above: since we use the iterative refinement method to design the relation between user characteristics and adaptation, we might early on get stuck in a design that will only cover certain of the problems that users needs to be aided in solving.

Another problem with scalability is the ability to take new user interests, preferences, knowledge, etc., into account. The same problem goes for when the system that the intelligent user interface is built on top of is redesigned, or, in intelligent information retrieval, when new kinds of information is entered, etc. In the area of information filtering, I believe that the solution lies in finding new combinations of human and machine intelligence in order to evaluate the relative importance of new information, [Höök et al. 1997]. In other areas, I think that good authoring tools that allows the designer to access the internals of the intelligent user interface without having to understand every detail of it, is needed.

So, the problem of scalability does not solve itself just because we use machine learning techniques. We still need to provide authoring tools that makes it possible to maintain, update and keep intelligent user interface's alive throughout the whole software development cycle – not only through the design phase. We must also consider the need for an interaction between human and machine intelligence in order to achieve high quality services that also can cover new areas.

6. SUMMARY

We have discussed four different challenges that we believe must be met before intelligent user interface's will be widespread: usability, development methods, useful adaptations, and maintainability. Some aspects of these problems must be investigated as the substantial research problems they are, and some may be solved if the intelligent user interface technology is properly transferred to industry.

These problems are not such that we should be discouraged to develop intelligent user interface's on these grounds, since there are a number of reasons why they will indeed be very useful. The widespread use of information technology means that the design of systems must meet many different user groups needs. Systems are getting to be more and more complex and users experience great difficulty with keeping up with the recurrent releases of software and the new possibilities offered. Another problem is the information overload that has come with the increased use of, for example, Internet. Yet another area where intelligence may be of help is in real-time critical applications. These are real problems not only to academics but also to all sorts of users around the world. We already have evidence that intelligent user interface's will be able to tackle some of these problems, we now need to focus our efforts in making it probable that the solutions rendered actually solve the problems they claim to solve.

REFERENCES

1. Benyon, David (1993) Adaptive Systems: A Solution to Usability Problems, Journal of User Modelling and User-Adapted Interaction, Kluwer, 3(1), pp. 1 -22.
2. Benyon, David R., and Murray, Diane (1993). *Developing Adaptive Systems to Fit Individual Aptitudes,* In W.D. Gray, W. E., Helfley and Murray, D. (eds.), Proceedings of the 1993 International Workshop on Intelligent User Interfaces, (pp. 115 – 122) Orlando, FL., New York, ACM Press.
3. Breuker, J. (ed.) (1990) EUROHELP: Developing Intelligent Help Systems, EC, Report on P280 ESPRIT Project EUROHELP.
4. Brusilovsky, P. and Pesin, L. (1994). *ISIS-Tutor: An adaptive hypertext learning environment* In H. Ueono & V. Stefanuk (eds.), Proceedings of JCKBSE'94, Japanese-CIS Symposium on knowledge-based software engineering, Tokyo: EIC.
5. Boyle, Craig, and Encarnacion, Antonio O. (1994). MetaDoc: An Adaptive Hypertext Reading System, *User Models and User Adapted Interaction*, (UMUAI) **4**, pp. 1 - 19.
6. Cole, Michael, (1996) Cultural Psychology: a once and future discipline, Harvard University Press.
7. Cook, R. and Kay, J. (1994) The Justified User Model: A Viewable, Explained User Model, Proceedings of the Fourth International Conference on User Modeling, Hyannis, Mass., The Mitre Corp.
8. Downs, E., Clare, P., and Coe, I. (1988) Structured Systems Analysis and Design Method: Application and Context, London, Prentice-Hall.
9. Horvitz, Eric (1997) Agents With Beliefs: Reflections on Bayesian Methods for User Modeling, Invited talk at Sixth International Conference on User Modeling, Chia Laguna, Sardinia, 2-5 June 1997.
10. Horvitz, Eric (1997) Compelling Intelligent User Interfaces: How Much AI is Enough?, Position statement, Proceedings of 1997 International Conference on Intelligent User Interfaces, (eds.) Johanna Moore, Ernest Edmonds, and Angel Puerta, ACM, Orlando, Florida, 1997.
11. Höök, K. (1996) A Glass Box Approach to Adaptive Hypermedia, Ph.D. Thesis, SICS Dissertation Series 23, ISBN: 91-7153-510-1, Stockholm, Sweden.

12. Höök, K. (1997) Evaluating the Utility and Usability of an Adaptive Hypermedia System, Proceedings of 1997 International Conference on Intelligent User Interfaces, (eds.) Johanna Moore, Ernest Edmonds, and Angel Puerta, ACM, Orlando, Florida, 1997.

13. Höök, K., Karlgren, J., Waern, A., Dahlbäck, N., Jansson, C-G., Karlgren, K. and Lemaire, B. (1996) A Glass Box Approach to Adaptive Hypermedia, Journal of User Modeling and User-Adaptive Interaction, UMUAI 6.

14. Höök, K., Rudström, Å., Waern, A. (1997) Edited Adaptive Hypermedia: Combining Human and Machine Intelligence to Achieve Filtered Information, presented at the Flexible Hypermedia workshop held in conjunction with the Hypertext conference, Southhampton, April, 1997.

15. Intelligent User Interface'97, International Conference on Intelligent User Interfaces Hilton at Walt Disney World Village, Orlando, Florida - January 6-9 1997.

16. Kaplan, Craig, Justine Fenwick and James Chen. (1993). Adaptive Hypertext Navigation Based On User Goals and Context, *User Modeling and User-Adapted Interaction* **3**, pages 193-220.

17. Kozierok, R. and Maes, P. A Learning Interface Agent for Scheduling Meetings, Proceedings of the ACM-SIGGHI International Workshop on Intelligent User Interfaces, pp. 81-88, New York, ACM Press, January 1993.

18. Kühme, T., Malinowski, U., and Foley, J.D. (1993). *Adaptive Prompting*, Technical Report GIT-GVU-93-05, Georgia Institute of Technology.

19. Lanier, Jaron, (1996) My problems with agents, Wired, 1996.

20. Lashkari, Y. , Metral, M., and Maes, P. Collaborative Interface Agents, Proceedings of the Twelfth National Conference on Artificial Intelligence, Vol. 1, AAAI Press, Seattle, WA, August 1994.

21. Lieberman, Henry, (1995) Letizia: An Agent That Assists Web Browsing Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, August 1995.

22. Maes, P. (1995) Intelligent Software: Programs that can act independently will ease the burdens that computers put on people, Scientific American, Vol. 273, No. 3, pp. 84-86, Scientific American, Inc., September 1995.

23. Maes, P. (1997) Direct Manipulation vs. Interface Agents, Position statement, Proceedings of 1997 International Conference on Intelligent User Interfaces, (eds.) Johanna Moore, Ernest Edmonds, and Angel Puerta, ACM, Orlando, Florida, 1997.

24. Meyer, B. (1994). Adaptive Performance Support: User Acceptance of a Self-Adapting System, Fourth International Conference on UM Hyannis, MA.

25. Nardi, B. (1996) Context and Consciousness: Activity Theory and Human-Computer Interaction, edited by Bonnie A. Nardi, The MIT Press, Cambridge, Massachusetts, London, England

26. Opperman, Reinhard (1994) Adaptively Supported Adaptability, International Journal of Human-Computer Studies, 40:455-472.

27. Orwant, J. (1994) Appraising the User of User Models: Doppelgänger's interface, In A. Kobsa and D. Litman (eds.), Proceedings of the 4thInternational Conference on User Modeling, pp. 73 - 78, Hyannis, Massachusetts, Mitre Corp.

28. Rich, Elaine (1979) User Modeling via Stereotypes, Cognitive Science 3, pp. 329 -354.

29. Recommender Systems, Communications of the ACM, March 1997 – Volume 40, no 3.

30. Roth, Emilie M., and Woods, David D. (1989). Cognitive Task Analysis: An Approach to Knowledge Acquisition for Intelligent System Design, G. Guida and C. Tasso (eds.) *"Topics in Expert*

System Design", Elsevier Science Publishers. B.V. (North-Holland), 1989.

31. Schank, R. (1991) Where's the AI?, AI Magazine, 12(4), pp. 38-48.
32. Shneiderman, B. (1997) Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces, Proceedings of 1997 International Conference on Intelligent User Interfaces, (eds.) Johanna Moore, Ernest Edmonds, and Angel Puerta, ACM, Orlando, Florida, 1997.
33. Sparck-Jones, Karen,
34. Suchman, L. A. (1987) Plans and Situated Actions: The problem of human-machine communication, Cambridge University Press.
35. Suchman, L. A. (1997) From Interactions to Integrations, Proc. of Human-Computer Interaction INTERACT'97, S. Howard, J. Hammond, and G. Lindegaard (eds.), Chapman & Hall.
36. Wærn, Annika (1996). Recognising Human Plans: Issues for Plan Recognition in Human-Computer Interaction, Ph.D. Thesis, ISBN 91-7153-456-3, SICS Dissertation Series 20, Stockholm, Sweden.
37. Wærn, Yvonne, Hägglund, Sture, Löwgren, Jonas, Rankin, Ivan, Sokolnicki, Tomas, and Steinemann, Anne, (1990). *Communication Knowledge for Knowledge Communication*, Research report, LiTH-IDA-R-90-17, Linköpings University, Sweden.
38. Wenger, Etienne (1987). *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*, Morgan Kaufmann Publ., Los Altos, CA.