Study of Framework for Mobile Interface

K. H. Walse
Associate Professor
Anuradha Engineering College,
Chikhli Distt. Buldana

R. V. Dharaskar
Director
MPGI Integrated Campus, Nanded

Dr. V. M. Thakare
Professor & Head
P.G. Dept. of Computer Science
S.G.B. Amravati University
Amravati

ABSTRACT
An adaphve user interface M supposed to adapt itself to the characteristics of an individual user. It is widely accepted that such an adaptation requires the interface to mamtain a user model embedded in the system

However, there are many unresolved problems with respeet to collecting reformation about the user and applying it in order to adapt the interface successfully.

Keywords

Adaptable, Adaptivity, Adpative Interfaces

1. INTRODUCTION
In the last few years a lot of work has been done in making systems more customizable or flexible. A flexible system increases the degree of freedom of usage, improves the correspondence between user, task, and system characteristics, and increases the user's efficiency. Two kinds of flexible systems are of special interest: adaptable and adaptive systems. A system is called adaptable if it provides the end user with tools that make it possible to change the system characteristics. It is called adaptive if it offers the ability to change its own characteristics automatically, possibly after consulting the user, thereby adapting itself to the user's needs.

Software systems developed in recent years are becoming increasingly powerful, but in most cases they tend to abandon the user to deal with the complexity of the system alone. There is an immense need for systems with individual, context-sensitive support[1]

2. RELATED WORK
The number of adaptable systems that are commercially available is on the increase, but so far these systems support very limited adaptation activities. For example, they may provide the user with tools for changing the user interface at hand. Making a system adaptable is reasonable, but this is only the fiist step in the development of more user-friendly systems. Adaptable systems are not sufficient, because they partly transfer the problem of designing a comfortable interface from the system developer to the end user. The designer of a system with a complex functionality has to make compromises in order to satisfy all possible needs (different users have different preferences and work styles, and one user may have different tasks at different times). In an adaptable system, the end user may override these compro- mises and tailor the system as he likes. He gets poor or no support in dealing with the customization features. There is a need for more context-sensitive system support, where the system knows more about its own tools and about the user, his work styles and tasks[1].

To give the user better support, the system must be able to analyze how the user interacts with the application and recognize when there is a problem. Unfortunately, systems seldom have built-in functions for this kind of evacuation. In order to reach the goat of more user-friendly systems, entirely new systems must be developed, or existing systems must be extended with the ability to anatyze the user's interaction and to offer individual support.

An adaptive system has knowledge about the system, its interface, the task domain and the user [2]. It must be able to match particular system responses with particular usage profiles.

In general, flexible systems may be scaled according to who makes the adaptation decisions, the system or the user. At one extreme are systems that are solely adaptable, i.e. the user alone is responsible for when and how to adapt. On the other extreme are systems that are solely adaptive, i.e. the system changes its characteristics without any consultation with the user. In between are solutions with shared decision making.

Each user may work differently with the adaptability of the system. The critique module can focus on the user's special needs and behavior concerning the adaptability. Such a module consists of a domain knowledge base with a set of rules and a usage profile. In our approach, the domain is "adaptability" and the usage profile describes the individual use of the adaptation possibilities.

In general, there are two ways to gain relevant information about the user. One way n a question-and-answer session requiring the user to provide self-estimations and exphclt preferences. Another way to obtain the necessary knowledge is to deduce it by monitoring the user's dialog with an application. Unfortunately, boti of these metheds present major problems. Self-estimations gwen by the user are not always reh able [1]. On the other hand, the deduction of reformation through dialog monitoring N most often severely restricted by a very small user-system communication bandwidth. Thus, the effort spent for an automated diagnosis of the user's behavior in dialogs M qwte high, compared to the usefulness of the assumptions obtained[2].

Applying the informahon gained and evaluating the success of adaptation is also very problematic. Users may be disturbed or confused by unexpected adaptations carried out automatically by the system. They might not feel in control of the system, what would be qtute in contrast to the ongmal retention of a system being dependent on its users. Furthermore, there is yet no generalized metric for a systematic evaluation of a performed adaptation [2]. In many situahons, users might be able to decide best on their own which N a successful adaptation and which is not. The control over adaptation should therefore be given to the users in order to enable them to make the required decisions. Control could always be

returned to the system at a user's command. On the other hand, there are adaptable systems which do give full control of adaptahon to the user. However, adaptations handled by the user are often restricted to a very low level. Achieving more than the simplest adaptations requires extra knowledge and an additional considerably large effort.

3. UBIQUITOUS COMPUTING

Three basic architectural design models for UbiCom system can be divided to smart devices, smart environment and smart interaction. The concept of "smart" means that the object is active, digital, networked, can operate autonomously, is reconfigurable and has a local control of the resources which it needs such as energy, data storage, etc

These three main types of system design may also contain sub-systems, sub-parts or components at a lower level of granularity that may also be considered as a smart (e.g., a smart environment device may contain smart sensors and a smart controller, etc). An example of a three main types of UbiCom models is presented in (Fig. 1) [4].

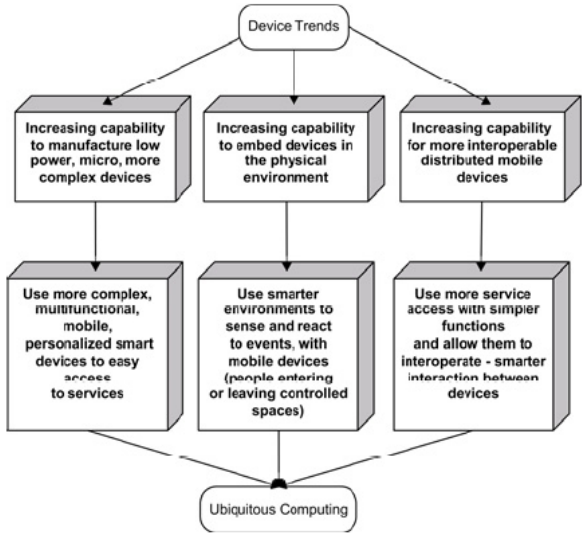

Fig. 1: Three models of ubiquitous computing: smart devices, smart environments and smart interaction [4]

Many sub-types of smarts for each of the three main types of smarts can be recognized. These main types of smart design also overlap between. Smart device can also support some type of smart interaction. Smart mobile device can be used for control of static embedded environment devices. Smart device can be used to support the virtual view points of smart personal spaces (physical environment) in a personal space which surrounding the user anywhere [4].

4. GUIDELINES FOR ADAPTIVE AND ADAPTABLE SYSTEM:

The most important guidelines were the following:

- For each adaptive feature, there must also be a corresponding adaptable one,

  Justification: Users must know that they are allowed to do at least everything that the system can do.

- There should be several ways of accessing the adaptation environment.

  Justification: Customization features are of little use if they are difficult to access,

- "At all times, the user should be in complete control of the system; the system may only act as assistant.

  Justification: System operation should be a creative process. Therefore, the user should not be forced into one specific working style.

- Suggestions from the system should not be "dramatic" and should not disturb the user unnecessarily in his work.

  Justification: System adaptation features are only aids to assist the user in getting the job done. Suggestions should not take the user's attention away from the real task.

- When possible, more than one adaptation possibility should be offered.

  Justification: A system is seldom able to spot the user's needs with 100~0 certainty. Adaptation suggestions should reflect this leaving freedom for the user to select between different adaptation possibilities.

- There must be an easy way to undo adaptations of the user interface. Additionally, there should be a simple way to reset all adaptations.

  Justification: The user interface should not be overloaded with adaptations which the user no longer needs or which have no relevance to the task at hand.

Having established these guidelines for adaptive behavior, the next step was to decide on a software platform on which to build the system.

We set up selection criteria, the most important ones being:

- "The system and the user interface must be complex enough for adaptivity to make some sense.
- The user actions must be recordable.
- The user interface must be modifiable.
- The system must have an up-to-date graphical user interface with which users are familiar, in order to make a realistic evaluation possible.
- "It must be possible to combine the original system with a knowledge base[1].

5. CONCLUSION AND FUTRURE WORK

We believe that the problem of providing adaptation is real and the solution process needs a more enginered process to be usable and cost effective. For the future we plan to further investigate this problem and to experiment our approach on further case studies.

6. REFERENCES

[1] Christoph G. Thorrtas & Mette Krogsczter, An Adaptive Environment For The User Interface Of Excel, , Intelligent User Interfaces '93, pp 123-130

[2] Thomas KGhme, A User-Centered Approach To Adaptive Interfaces, Intelligent User Interfaces '93, pp 243-245

[3] Luca Cavallaro and Elisabetta Di Nitto, An Approach to Adapt Service Requests to Actual Service Interfaces, SEAMS'08, May 12–13, 2008, Leipzig, German

[4] Ondrej Krejcar, Complex Mobile User Adaptive System Framework For Mobile Wireless Devices, un published work

,