Helping users through ubiquitous, personalised, interactive support in a sightseeing visit

Luisa Marucci, Fabio Paternò
CNUCE-CNR
Via G. Moruzzi, 1
Pisa, Italy
+39 050 3153066
{luisa.marucci, fabio.paterno}@cnuce.cnr.it

ABSTRACT
This paper introduces an approach to providing intelligent support through different devices to organize and make a generic sightseeing visit involving indoor (a Museum) and outdoor areas (the city historical centre). The goal is to provide users with access to a range of information via wired and wireless client devices. The support is based on user modelling techniques that can update information at run-time about user preferences and knowledge expressed through any device used to make the visit. Such information provides the basis for tailoring the visit to the specific user, taking into account the accesses through different interaction devices.

Keywords
Adaptivity, User model, Task Model, Heterogeneous interaction platforms.

INTRODUCTION
Applications that can be accessed from a variety of different people (with different backgrounds, cultures, aims) and characterized by broad information domains, call for support in order to allow users to best use the system. A good example is museum applications.

When we are in a museum or, more generally at an exhibition, it is important to have support (for example a guide) that takes into account what we have already seen and the paths we have chosen, in order to help us better understand the sights we are viewing.

In addition, the increasing availability of many types of devices for accessing the same application raises a number of challenges to user interface designers. There is a need for interactive applications able to adapt plastically (Calvary, Coutaz, Thevenin, 2001) also to the different contexts of use. We consider the context of use to include the types of devices that support users while performing their tasks and the surrounding environment.

To address these issues, we propose a single underlying user model available when users interact via different devices, which is used to provide them with assistance during ubiquitous access to the information regarding artworks.

Depending on the platform currently being used and on the preferences and knowledge-level extracted from the past user sessions, the virtual assistant decides what to show to the user (what information, which links to suggest and so on). This is different from the support proposed in (Ardissono, Goy, Petrone, Segnan, Torasso, 2002), where users can access and manipulate the same data through different platforms. Here, we also want to change the way the user interface is presented and supports navigation, by accounting for the interactions performed through the various devices.

User modelling (Brusilovsky, 1996) is an approach introduced to support adaptive interfaces that change some of their parts according to user interactions. It can also be helpful when applications accessible through multiple interaction devices are considered. User models aim to represent aspects regarding users, such as their knowledge level, preferences, background, goals, position, etc. Such information is useful to furnish user interfaces with adaptivity, that is, the ability to dynamically change their presentation, content and navigation, in order to better support users' navigation and learning, also considering the current context of use. Several representations of user models have been used: concepts related to the users' knowledge and values indicating their importance for their goals; user stereotypes and probabilities indicating their relevance to the current user. Various aspects of the user interfaces can be adapted according to user models. They can adapt their text presentation through techniques such as conditional text or stretch-text. They can also adapt the user navigation using techniques such as direct guidance, adaptive order of links, hiding of links. Adaptive techniques have been applied in many domains. The example presented in (Marucci, Paternò, 2001) describes a Web system supporting an adaptive museum guide that provides virtual visitors with different types of information (introduction, summary, comparison, difference, curiosity) according to their profile, knowledge level, preferences, and history of interactions.

To date, only a few works have considered user modelling to support the design of applications accessible through different devices, and there are still many issues that need to be solved in this context. An example is Hippie (Opperman, Specht, 2000) a prototype that applies user modelling techniques to aid users in accessing museum information through either a web site or a PDA while in the museum. In our case we address the use of mobile out-door technologies and provide user models integrated with task models developed at design time.

In this paper we present an approach that shows how user modelling can be leveraged to support users accessing an application through multiple interaction devices. The basic idea is to have a single user model associated with each user that is dynamically updated and used when the user interacts with the application through any type of device (see Figure 1). We discuss our approach using a case study in the museum application domain. This application area has inspired a number of works regarding design of applications for mobile devices. For example, an interesting work in this area is (Aoki, Woodruff, 2000), which cautions that user location should not be considered an absolute indicator of interest in the closest artwork. Outdoor support for tourists is explored in (Cheverst, Davies, Mitchell, Friday, Efstratiou, 2000), which has limited adaptive support.

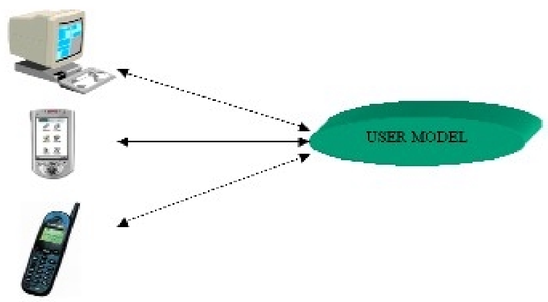

Figure 1: User Model support for heterogeneous platform.

In our case the adaptive support provided takes into account both the previous user interactions and the device type with which they were performed. Examples of possible adaptivity are:

- A visit is started with a desktop computer and continued with a cellular phone: when users log in through the mobile phone the application is able to inherit their preferences and knowledge levels (derived from the previous interactions with the desktop version), and present lists and parts of information in an adaptive manner taking into account previous accesses (for example, lists of links are ordered in terms of importance for the current users and less important links are not immediately displayed);
- In the case of a tourist visiting a town, the possibility of selecting works of art that are deemed particularly important and annotating them via a mobile phone and, then, when the application is accessed through a desktop system, it will immediately display more detailed information on the selected works of art;
- The possibility of automatically detecting the route followed during a physical town visit, which the application is then able to propose in a virtual manner through the desktop system.

In the paper we first introduce a detailed scenario to highlight the type of support that we have designed. Then, we discuss the types of rules that drive the adaptive behaviour exploiting the information in the user model. We also describe our method, how the user model is structured and how such information is used to obtain the adaptive user interface. Lastly, we provide some concluding remarks.

EXAMPLE SCENARIO

John starts to visit the Web site providing information about Carrara from the hotel with a desktop computer. He finds it interesting. In particular, he is interested in marble sculptures located close to Piazza Garibaldi. He spends most of his time during the virtual visit (Fig.2 – part a) accessing the related pages and asking for all the available details concerning such works of art.

Figure 2: Spatial information provided through the desktop (a) and the phone (b).

The day after he leaves the hotel and goes to visit the historic town centre. When he arrives he accesses the map of the town through his personal login via phone (Fig.2 – part b).

The system inherits his preferences and levels of knowledge from the virtual visits performed in the hotel. Thus, it allows him to access information on the part of the town that prompted his interest most, and navigation is supported through adaptive lists based on a ranking determined by the interests shown in the previous visit through the desktop system. During the physical visit he sees many works of art that impress him, but there is no information available nearby, so he annotates them through the phone.

When he is back in the hotel, in the evening, he again accesses the town web site through his login. The application allows him to access (Figure 3) an automatically generated guided tour of the town following an itinerary based on the locations of the works of art that impressed him (Figure 4). He can modify it if he no longer finds interesting some of the proposed works of art. So, he can perform a new visit of the most interesting works of art receiving detailed information regarding them.

Figure 3: The User Interface to the desktop version after an access through the phone version.

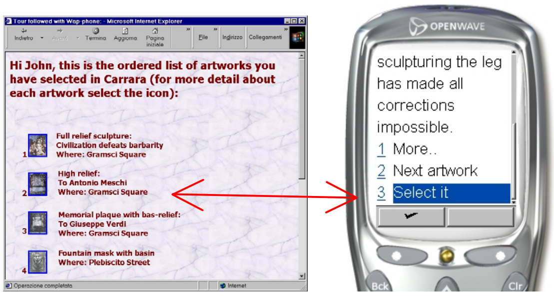

Figure 4: On the left side the desktop interface to access the artworks selected with the phone (right side).

CRITERIA FOR ADAPTIVE USER INTERFACES

This section describes the rules that are used to drive the adaptivity of the user interface. We will explain how they are handled, highlighting the resulting adaptive navigation and presentation modality consequent to the users' interactions with the system through different platforms. The following scheme shows when a rule comes into force and how the interactive system behaviour changes accordingly.

Adaptive Navigation Modality	
When: The user always performs the same sequence of tasks that leads to accessing the same domain object	**How**: Addition of an interface element in each platform that points directly to the domain object selected
When: The user performs a specific task in one platform and then accesses the application through the other platform	**How**: Enable or disable some tasks in the other platform
When: The user never selects a task (for example, a link selection) during one or more sessions (in any platform)	**How**: Removing the task support from each platform (for example, remove link)

Adaptive Presentation Modality	
When: The user often selects a domain object subset (independently of the task order and the platform)	**How**: Ordering each list that contains this object accordingly
When: The user never selects a domain object or an attribute	**How**: Putting access to this object or attribute in a non-priority position

Changing the Navigation Modality according to task frequency

Here we consider the more general rule in order to show how the system handles such situations (the user always repeats the same sequence of tasks). For example, we can consider when the user selects a subset of domain objects and then a more refined subset iteratively. Thus, we can consider a sequence of tasks that involve a two-stage selection: first the user chooses one general topic and then a subtopic.

The recurrent selection of a specific type of artwork (made of *bronze*, defined as *full relief sculpture*, etc.), followed by a more specific selection (*bronze* artworks from the *XX Century*, *full relief sculpture* by the artist *Vatteroni*, etc.) implies the appearance in the interface of a new link to allow the user to directly access the subclass selected: "*Bronze artworks in XX Century*" or "*Vatteroni's full relief sculpture*".

Such link will appear until the user has visited all the artworks belonging to that subset and/or until the system detects different preferences.

We can follow the corresponding changes in the user model: for each task there is an attribute which represents the possibility of that task's being "*merged*", an indication of the task to which it can be connected and the new name to be given to this unified task as well as the number of accesses, object instance and object subset selected.

In the foregoing example this will generate a link "*Bronze artworks in XX Cen.*", in both the desktop and mobile interfaces in which the user can select the material.

The number of additional links depends on the type of platform. During dynamic generation of the user interface, the system first analyses the user model and then consequently generates the links.

Another example is when the user never performs some tasks during a session or during different sessions. In this case the system will remove the tasks in question. Thus, the task can be disabled (by setting the corresponding attribute) if it is never performed (over one or more sessions) in any platform (presuming the task is defined for multiple platforms).

Changing the Navigation Modality according to task dependencies

Here we consider tasks performed in a specific platform which generate a change in task model related to another platform. For example, the user previously selects a tour with the desktop, indicating preferences for a city zone and then accesses the application through the cellular phone. When the user selects a tour through the desktop, via either map access or the predefined link, the task "*Follow the desktop selected route*" in the user model will be modified: the corresponding *Disabled* attribute, previously set to true, will be set to false, and the corresponding object instance will be the tour chosen by the user.

Figure 5: Access to the application for the first time and after desktop visit and tour selected.

Vice versa, from the phone platform the user can choose the option of selecting the same artworks encountered during the visit in order to see them better with the desktop. This will enable the task "*More Information about artworks visited*" in the desktop platform, and each of the artworks selected will be added as the objects corresponding to that task.

Changing of Presentation Modality according to the objects selected

We now show an example of a changing presentation considering a task whose related objects are the artworks located in the historical city centre of Carrara.

The user can access such artworks by choosing one of the alternatives: Streets, Buildings, Churches, or Squares. Suppose that the user often chooses "Streets".

The user model contains the objects corresponding to the types of artworks in Carrara city (and the specific platforms from which each of them can be accessed).

More generally, the user model also contains indications of the objects manipulated by each task as well as the platforms supporting each object.

For each user choice, the system stores the results of the selections in the user model. In the example mentioned above, the user first selects "artworks in Carrara city" and then the object "Streets".

The recurrent choice of this attribute will determine a change in sorting of the items in the corresponding list (see Figure 6).

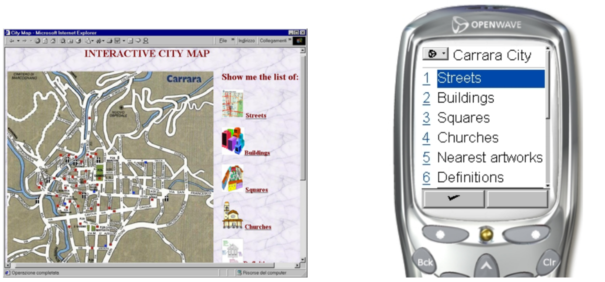

Figure 6: Example of adaptive lists.

To sum up, if the user selects an object while performing a task through one platform and that same object is associated with a task performable through other platforms, this will determine a change in the sorting of each list containing the selected object, regardless of the platform.

Changing the Content Presentation

In this case we consider user interactions that modify the user's knowledge level. This generates a modification of the content presentation (while maintaining the same navigation modality).

An example of this rule is when the user accesses the description of an artwork. The level of detail depends on the user knowledge of the application domain (Figure 7).

Figure 7: The User asks frequently for more information, the system automatically generate it.

While the user interact with the application, the knowledge level of the user is updated. After a certain number of accesses the level of knowledge is automatically increased and the system provides more detailed information. When the user accesses the system from any platform the knowledge level will be inherited.

UNDERLYING SUPPORT

In our approach we assume that a model-based method has been followed to the design of the multi-platform application. Recent developments of the ConcurTaskTrees notation (Paternò, 1999) allow designers to develop task models of nomadic applications. This means that in the same model designers can describe tasks able to be performed on different platforms and their mutual relationships (Paternò, Santoro, 2002).

From this high level description it is possible to obtain first the system task model associated with each platform and then the corresponding user interface. The task model can be represented in two ways: a graphical representation that can be edited and analysed with the CTTE tool (publicly available at http://giove.cnuce.cnr.it/ctte.html) or in XML format that can be automatically generated.

The user and task models share some information, but also contain different elements. This means that some elements of the task model are removed and others added (such as the association for each task of the attributes for the adaptive navigation support). In addition, the user model is mainly characterised by values that are updated dynamically according to the interactions performed by the corresponding users. For each user we have a user model that is updated when the user interacts with any of the available platform. The contents of the user model are used by a run-time support that accordingly modifies the user interface presentation, navigation and content by applying some previously defined adaptivity rules.

One advantage of this approach is that the task model developed at design time already provides some useful information for the run-time adaptive support: the temporal dependencies among tasks performed on different platforms, the tasks executable from many platforms, the association of tasks with domain objects and the related attributes (as well as the definition of objects and attributes accessible through a specific platform).

The performance of some tasks (from either phone or desktop) can change the level of interest associated with some domain objects (for example the preferred city zone), and this information can also be used to adapt the presentation support for a platform different from that currently in use (for example, the order of the links in a list).

The user model contains information such as the number of times a task has been performed or an object has been accessed. It also contains fields that allow dynamic modification of the task availability: *Mergable* indicates whether it is possible to enable the task along with a different task at the same abstraction level, *Hidable* indicates whether it is possible to disable its performance including it in another, more general, task, and *Disabled* whether it is possible to completely disable it for the current user.

In figure 8 we can see an excerpt of task model developed using the CTTE tool:

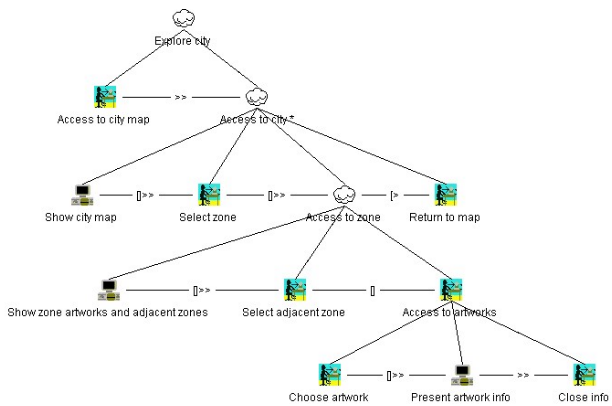

Figure 8: An example of task model.

In addition, for each task all the attributes are defined (through the tool) including the properties related to the adaptive support, and at the end the tool generates the xml file that will be used to generate the user model.

The information in the user model is updated dynamically according to the user interactions.

From analysis of the user model, the system is able to determine the tasks performed by the user and in what order, as well as the objects classes and related subclasses chosen. From this user input then, the navigation preferences will be detected (by analysing the sequence of tasks chosen, the tasks never performed, the task usually performed, etc.), as will the presentation preferences (by analysing the objects classes and objects subclasses accessed).

The location is an attribute related only to mobile interactive platforms, for example after a user has selected an item from the *Materials* list, then the system provides the possibility of selecting only the artworks made of such material located nearby (for example, with users with WAP phones).

The domain model is structured in terms of objects classes and the related subclasses that are manipulated during task performance. The relationships between tasks and domain objects are represented in the user model. The association between tasks and object instances can be either static or dynamic. For example, in the task of selecting an element from a list of predefined values, the association is static, whereas the association is dynamic in the task of presenting information on a work of art (whose name is provided by the user).

The domain objects that can be accessed and manipulated vary according to the device that is available. In general, the domain objects that can be manipulated via phone are more limited than those accessible via desktop computers and have some different spatial attributes related to the user position, such as the *closeness*.

Likewise, the supported tasks depend on the interaction platform: there are tasks associated with a desktop virtual visit and others associated with the phone-supported visit, but performance of some kinds of tasks on one platform may depend on the accomplishment of other tasks through other devices (for example the desktop task associated with reviewing the itinerary annotated by phone).

The user model is dynamically updated to indicate knowledge and preferences of the user in terms of objects and attributes and the performance frequency of each task depending on the user behaviour.

The navigation enabled by the application can be dynamically changed according to the user interactions that reveal frequently or never performed tasks or preferences in terms of interaction techniques.

Conclusions

In the paper we have discussed how adaptive support based on user modelling techniques can be provided when interactions through multiple platforms are considered. We have shown the type of design that can be obtained through a case study in the cultural heritage application domain.

In particular, we have discussed a set of rules that make it is possible to change the presentations and dialogues supported by the user interface by taking into account users' interactions through different platforms. This results in greater application flexibility.

Future work will be dedicated to evaluating in depth the usability of adaptive support, in particular through small devices where, for example, changes to the list of links may result more confusing than on desktop systems, if they are not carefully introduced.

REFERENCES

Aoki P. M., Woodruff A., "Improving Electronic Guidebook Interfaces Using a Task-Oriented Design Approach", 3rd Conf. On Designing Interactive Systems, NY, Aug. 2000, pp. 319-325

Ardissono L., Goy A., Petrone G., Segnan M., Torasso P., "Ubiquitous User assistance in a Tourist Information Server", Proceedings Adaptive Hypermedia 2002, LNCS 2347, pp.14-23.

Brusilovsky P., "Methods and techniques of adaptive hypermedia, User Modelling and User Adapted Interaction, v. 6, n. 2-3, pp. 87-129, 1996 http://www.cntrib.andrew.cmu.edu/plb/UMUAI.ps

Calvary G., Coutaz J., Thevenin D., "A Unifying Reference Framework for the Development of Plastic User Interfaces. Proceedings of Engineering HCI, Toronto, Canada May 11-13, 2001

Cheverst K., Davies N., Mitchell K., Friday A., Efstratiou C., "Developing a context-aware electronic tourist guide: some issues and experiences, Proceedings ACM CHI 2000, pp. 17-24

Marucci L., Paternò F., "Design and Evaluation of an Adaptive Virtual Guide for Web Applications", Universal Access in the Information Society, 2001, Springer Verlag

Opperman R., Specht M., "A Context-sensitive Nomadic Information System as an Exhibition Guide", Proceedings of the Handheld and Ubiquitous Computing Second International Symposium, pp. 127-142, HUC 2000, Bristol, UK, September 25-27, 2000.

Paternò F., "Model-based Design and Evaluation of Interactive Applications", Springer Verlag, ISBN 1-85233-155-0, 1999.

Paternò F., Santoro C., "One Model, Many Interfaces", Proceeding CADUI 2002, pp.143-154., Kluwer 2002.