# Supporting Adaptivity with Heterogeneous Platforms Through User Models

Luisa Marucci, Fabio Paternò

CNUCE-CNR
Via G.Moruzzi 1
Pisa, Italy
Fax: +39 050 3138091
luisa.marucci@guest.cnuce.cnr.it
fabio.paterno@cnuce.cnr.it

**Abstract.** In this paper we describe an approach to providing adaptive support to multiplatform applications, which can be accessed through multiple interactive devices from various locations. It is based on task-models including context-dependent information and a single user model, which can update information about user preferences and knowledge at run-time. Such information is used to adapt the navigation, presentation and content of each user interface also taking into account users' accesses through different interaction platforms.

## 1 Introduction

The increasing availability of many types of devices and the next generation mobile technologies (such as UMTS), wireless LAN-based solutions, and new type of terminal equipment (such as wearable computers) raise a number of challenges to user interface designers. To this end there is a need for interactive applications able to adapt in a plastic manner [2] to the different contexts of use. We consider that the context of use includes the types of devices that support users while performing their tasks and the surrounding environment.

User modelling [1] is an approach that aims to represent aspects regarding users, such as their knowledge level, preferences, 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. 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 and also the kind of presentation from text to speech or vice versa. They can also adapt the user navigation using techniques such as direct guidance, adaptive order of links, hiding of links.

To date, only a few works have considered user modelling to support the design of multi-platform applications. An example is Hippie [3] 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 the use of mobile out-door technologies and user models integrated with task models developed at design time is also considered.

This paper presents an approach that shows how user modelling can be leveraged to support users accessing an application through multiple interaction devices and the definition of adaptive techniques that takes into account also the device used. The basic idea is to have a single user model associated with each user that is dynamically updated when the user interacts with the application through any type of device. We discuss our approach using a case study in the museum application domain.

### 1.1 The method

This approach assumes that a model-based method has been followed to the design of the multi-platform application. Recent developments of the ConcurTaskTrees notation [4] allow designers to develop task models of multiplatform applications. This means that in the same model, designers can describe tasks able to be performed on different platforms, their mutual relationships and what platforms are suitable for each task.

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 a tool (publicly available at http://giove.cnuce.cnr.it/ctte.html) or in XML format that can be automatically generated.

In our case we use the XML specification as input for the creation of the user model. In addition, the user model is mainly characterized by values that are updated dynamically according to the interactions performed by the corresponding user with any of the available platform. These values are used by a run-time support that modifies the user interface presentation, navigation and content according to 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).

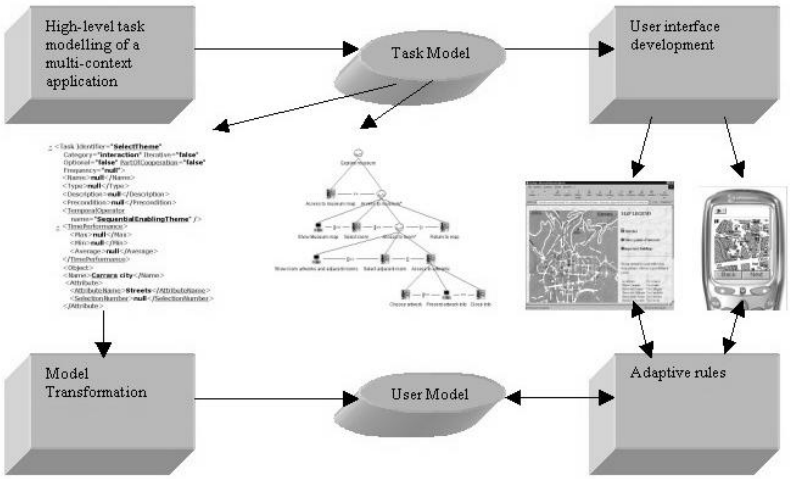


**Fig. 1.** The method proposed

The navigation preferences will be detected by analysing the sequence of tasks chosen, the tasks never performed, the task usually performed and so on, as will the presentation preferences (by analysing the objects classes and objects subclasses accessed).

The user model also contains fields that allow dynamic modification of the task availability, for example whether it is possible to merge more tasks at the same abstraction level in one, whether it is possible to disable a task performance including it in another, more general, task, and whether it is possible to completely disable a task for the current user.

Likewise, the supported tasks depend on the interaction platform: there are tasks associated with a desktop virtual visit, 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).

Also 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. In addiction there are some spatial attributes related to the user position that are meaningful only for mobile devices.

### 1.2 Rules and algorithms

This section describes examples of the adaptive rules that are used to drive the adaptivity of the user interface. We explain how they are handled, highlighting the resulting adaptive navigation and presentation modality consequent to the users'

interactions with the system through different platforms. In succession, we show *when* a rule comes into force and *how* the interactive system behavior or presentation 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 other platform | **How**: Modifying the task model state of the other platform to enable or disable some tasks |
| **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 |

**Table 1.** Examples of Adaptive rules

### 1.3 An example: changing the Navigation Modality according to task dependencies

Here now an example of tasks performed in a specific platform which generate a change in task model related to another platform (rule 2): the user previously selects a tour with desktop, indicates preferences for a city zone and then accesses the application through the cellular phone (see Figure 2).


**Fig. 2.** Access to the application for the first time, after desktop visit and tour selected and after desktop visit and no tour selected

Vice versa, from the wap platform the user chooses the option of selecting the same artworks encountered during the visit in order to see them better with 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 and visualized on user request (figure 3).

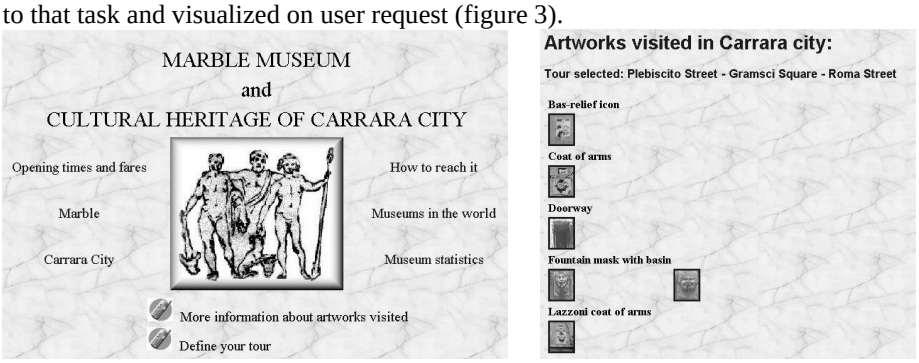


**Fig. 3.** The user interface to the desktop version after an access through the phone version

## 2 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 museum 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 usability.

## References

1. Brusilovsky, P.: Methods and techniques of adaptive hypermedia, User Modelling and User Adapted Interaction, v 6, n 2-3, pp.87-129, 1996 URL: http://www.cntrib.andrew.cmu.edu/plb/UMUAI.ps
2. Calvary, G., Coutaz, J., and Thevenin, D.: A Unifying Reference Framework for the Development of Plastic User Interfaces. Proceedings of Engineering HCI, Toronto Canada May 11-13, 2001
3. Oppermann, R., Specht, M.: A Context-sensitive Nomadic Information System as an Exhibition Guide. Proceedings of the Handheld and Ubiquitous Computing Second International Symposium, HUC 2000 Bristol, UK September 25-27 2000, pp.127-142
4. Paternò, F.: Model-based Design and Evaluation of Interactive Applications, Springer Verlag, ISBN 1-85233-155-0, 1999