Ad-Hoc Interactions in a Museum

Fabio Paternò, Carmine Ciavarella, Luisa Marucci, Carmen Santoro

CNUCE - C.N.R.
Pisa, Italy
fabio.paterno@cnuce.cnr.it

Abstract. Recent years have seen an increasing availability of interaction device types. This has posed a number of issues for identifying suitable criteria for the design of interactive software applications. At the same time, network communication technology has evolved enabling more flexible protocols for communication among dynamic sets of devices, thus opening up further possibilities. In this position paper we discuss the various levels of interactive support that is possible to make available using a museum application as case study. The aim is to enable each interaction device to support the appropriate tasks users expect to perform and designers to develop the various device-specific application modules in a consistent manner.

1. Introduction

Recent years have seen the introduction of many types of computers and devices (e.g. cellphones, PDA's, WebTV, etc.) and the availability of such a wide range of devices has become a fundamental challenge for designers of interactive software systems. Users wish to be able to seamlessly access information and services regardless of the device they are using, even when the system or the environment changes dynamically. To this end, computer-based applications need to run on a wide spectrum of devices.

Designing applications that exploit new technology is often a difficult problem. For software developers this introduces the problem of constructing multiple versions of single applications and endowing these versions with the ability to dynamically respond to changes in context. Creating different versions of applications for different devices engenders extra development and expensive maintenance cost of cross-platform consistency, complicates the problems of configuration management and dilutes the resources available for usability engineering. In addition, current development tools provide little support for creating applications that change dynamically in response to changes in their environment, or that have to share data amongst heterogeneous device types.

Many dimensions must be considered when designing context-dependent applications (actors, platforms, environments, system resources, etc.). In the CAMELEON project (**http://giove.cnuce.cnr.it/cameleon.html**) we have developed a method to support design and development of highly usable context-sensitive interactive software systems. Such a method has been applied to the development of a multi-platform nomadic application for accessing information regarding a museum and the historical centre of the town of Carrara (Italy). To this end, the nomadic application is composed of a Web site of the Marble

Museum supporting access to hundreds of works of art, a PDA support implemented in an IPAQ Compaq PDA already in use for the museum visitors, and a prototype of WAP application. We are interested in understanding how to introduce and support ad-hoc user interactions in a new version of this museum application and the benefits that they can provide for the visitors.

More generally, interest on how to introduce mobile interactive devices to support museum visitors (see for example [1] [3] [5]) has been increasing in recent years. However, the possibilities opened up by ad-hoc networks have not yet been carefully investigated for this application domain.

The evolution of this research line is useful to highlight the design space that recent technological devices and infrastructures are opening up.

We identify three-level of interactive support. Their main features are:

- *Current support*: *single user* interacting with the nomadic application through *multiple interaction platforms* (cellular, PDA, desktop system) *at different time*, supporting *predefined sets of tasks*;
- *Single user ad hoc support*: *single user* interacting with the nomadic application through *multiple interaction platforms* (cellular, PDA, desktop system, intelligent boards, interactive artefacts) *at the same time* with *run-time dynamic allocation of tasks depending on devices' proximity with the* support of IEEE 800.11, Infrareds and Bluetooth technology;
- *Multiple users ad hoc support*: *multiple users* interacting with the nomadic application through different interaction platforms (*cellular, PDA, desktop system, intelligent boards, interactive artefacts*) *at the same time* with *run-time dynamic allocation of tasks depending on proximity of devices and users* with the support of IEEE 800.11, Infrareds and Bluetooth technology.

One technique that we consider at each possible level of interactive support is the use of user modelling [2] that allow us to have dynamic information regarding user preferences and knowledge that can be used at run-time for adapting various aspects of the user interface (navigation, presentation, content).

In our approach the user model is structured in such a way as to indicate user preferences (for example, the preferred city zone, navigation style, theme or features of an artwork) and acquired knowledge (for example, the level of knowledge about an author, an historical period or a material) depending on the user's accesses to the application.

The user model contains information that is dynamically updated such as the number of times a task has been performed or an object has been accessed in any of the platform available (see Figure 1). 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.

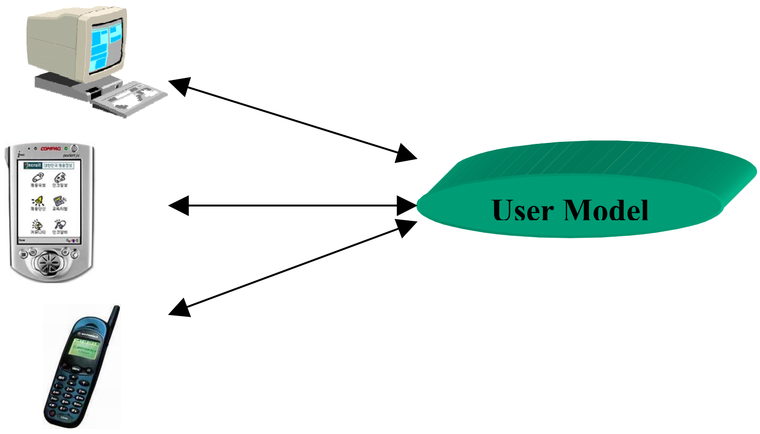

Figure 1: User model supporting interaction through multiple devices.

In the next sections we first introduce our model-based method for the design of nomadic applications and then we discuss some scenarios to better highlight our approach to these levels of support and what implications they have in terms of design.

2 Model-based design of nomadic applications

The design of multi-platform applications has to consider a number of issues. It is not possible to support the same type of tasks with different devices just changing the set of interaction and presentation techniques according to the resources available in the device considered. It is important that designers consider different devices also with regard to the choice of the tasks to support (e.g. phones more suitable for quick access to limited information, desktop systems more suitable for browsing through large amounts of information). To complicate matters, even within the same class of devices there are different presentation models that need to be handled: for example, in WAP-enabled phones a number of micro-browsers tend to accept slightly different versions of WML, must interact with slightly different phones (for examples, phones with a different number of softkeys) and interpret the softkey interactions differently.
Our method tries to address such problems, and is composed of a number of steps that allows designers to start with an overall envisioned task model of a nomadic application and then derive concrete and effective user interfaces for multiple devices:

- *High-level task modelling of a multi-context application*. In this phase designers need to think about the logical activities that have to be supported and relationships among them. They address the various possible contexts of use and the various roles involved and also a domain model aiming to identify all the objects that have to be manipulated to perform tasks and the relationships among such objects.

- *Developing the system task model for the different platforms considered.* Here designers have to filter the task model according to the target platform. This involves identifying task models in which the tasks that cannot be supported in a given platform are removed and the navigational tasks deemed necessary to interact with the considered platform are added. Such models can be specified using the ConcurTaskTrees notation. The CTTE (CTT Environment) tool (publicly available at http://giove.cnuce.cnr.it/ctte.html) supports editing and analysis of task models specified using this notation.
- *From system task model to abstract user interface.* Here the goal is to obtain an abstract description of the user interface composed of a set of abstract presentations that are identified with the support of the enabled task sets and structured by means of various operators. Then, we identify the possible transitions among the user interface presentations considering the temporal relationships that the task model indicates.
- *User interface generation.* In this phase we have the generation of the user interface. This phase is completely platform-dependent and has to consider the specific properties of the target device. For example, if the considered device is a cellular phone, such information is not sufficient as we also need to know the type of micro-browser supported and the number and the types of soft-keys available.

3. Current nomadic application 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 Alberica. He spends most of the time of the virtual visit in accessing the related pages and asking for all the available details concerning such works of art. The day after he leaves the hotel and goes to visit the historical centre of the town. When he reaches it, he accesses the Wap site of the town through his personal login. The system inherits his preferences and levels of knowledge after the virtual visits performed in the hotel. Thus, it starts providing information on the part of the town that prompted his interest most and the navigation is supported through adaptive lists based on a ranking determined by the interests shown in the previous visit through the desktop system (see Figure 2). During the physical visit he sees many works of art that impress him but there is no information available regarding them so he annotates them through the Wap phone. When he is back in the hotel, in the evening, he accesses again the town web site through his login. The application automatically generates a guided tour to the town following an itinerary based on the locations of the works of art that impressed him. So, he can perform a new visit of the most interesting works of art receiving detailed information regarding them.

Figure 2: Access to the application for the first time, after desktop visit and tour selected and after desktop visit but no tour selected.

4. Ad-hoc support for single user scenario

Piero is visiting the Marble Museum with his PDA (see Figure 3). He has received headphones to better hear information provided through the audio channel. In the first room he finds a number of showcases containing a number of small models and characters related to marble quarrying. As soon as he is close to them they start to animate showing how marble is quarried and through the headphones he can hear typical noises and voices that create a more realistic atmosphere. The PDA shows the list of characters in the model. He selects one of them and then he receives more information regarding his role. In addition, through the PDA he can control his behaviour.

Figure 3: PDA support when the user gets in a room.

In the next room, there is a wide plastic model. He can pass close to it. A small interactive kiosk is located close to the most meaningful part. When he is close to the kiosk, it starts showing videos and interviews with famous artists.
During the visit he moves to a new room. As soon as he gets in a video starts. The video concerns the geographical area related to the theme of the exhibition. Through his PDA Piero can control the video projector, for example to change light, sound, brightness (see Figure 4). Piero can also directly interact with the video. For example, he can select the area of interest by touching it on the wall where the video is shown. When a new user gets in the room and wants to interact with the video through the wall then Piero's projection is continued in his PDA.

Figure 4: Dynamic change of PDA-supported task.

5. Ad-hoc support for multiple users scenario

A museum visit involves a group of heterogeneous people: art students, professors and accompanying people.
When they arrive at the museum, a guide implemented in PDA is given to each person: such PDA has audio and video capabilities. Each user can enter some preferences into the PDA, so a customised presentation can be supplied within the PDA. The PDA shows the list of events scheduled for the day in the town, together suggestions for a customised agenda and route through the museum depending on the selected profile. Alternatively, the user can select the events in which he is interested in order to enable the PDA to create a customised agenda depending on such parameters. When an event is going to start a warning is activated on the PDA.

When this heterogeneous group enters the museum they decide to divide in smaller groups to have the possibility of visiting the museum freely. Also cooperative features are enabled within the PDA for example to allow people decide how to meet each other within the museum or highlight interesting elements to other people who are in other locations.

Within the museum a number of works of arts are shown with also the possibility of dynamically interact with some intelligent artefacts (see previous scenario). When a character is under control of one visitor it is unavailable for the others. If a new user arrives then he can control one of the remaining available characters. The characters can interact with each other in order to show how people cooperate when quarrying marble. The type of control that a user can perform depends on his knowledge of the application domain. The assumption is that visitors with a good knowledge of marble quarrying (such as the professors) have a better idea of the roles associated with each type of character and consequently they have clearer ideas of what they can do. Such level of knowledge is indicated through the PDA at the beginning of the visit and his dynamically updated according to user interactions.

In the museum there are rooms with ongoing exhibitions and also rooms where it is possible to interview artists: such rooms are connected in videoconference so it is possible to send questions to artists although you are not in the room: the questions can be recorded even with a cellular phone and then sent and projected on the screen that is in the room.

Within the PDA museum maps are displayed with the possibility of annotating them with comments and they also show where the other people of the group are at that time.

6. Discussion

These scenarios highlight how ad-hoc features open up interesting possibilities in terms of flexible interactions able to adapt to different configurations of devices, people and context. However, our first experiences in this area show that attention should be paid to design how these adaptive features are made available to users. There is a risk of disorienting users in changing suddenly the available interactions and functionality. Thus, these changes should be smooth and use consistent rules and occur as the natural consequence of changes of context. Adaptive techniques are flexible and able to overcome some limitations of previous approaches. However, a completely automatic adaptive support can be rather confusing for end users, who may not understand the reasons for the dynamic changes occurring in the system's behaviour. Thus, when designing adaptive support, it is important to allow users to clearly understand:

- When the adaptive support can be activated;
- How the adaptive support provides information;
- Which criteria determine the generation of information provided by the adaptive support.

Acknowledgments

CAMELEON is an IST R&D European project. More information is available at http://giove.cnuce.cnr.it/cameleon.html

References

[1] P. M. Aoki, A. Woodruff, Improving Electronic Guidebook Interfaces Using a Task-Oriented Design Approach, *3rd Conf. on Designing Interactive Systems*, New York, NY, Aug. 2000, pp. 319-325

[2] 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

[3] C.Ciavarella, F.Paternò, Design of a Handheld Interactive Support, CNUCE Internal Report, November 2001.

[4] L.Marucci, F.Paternò, Supporting Interactions with Heterogeneous Platforms Through User Models, CNUCE Internal Report, February 2002.

[5] R.Oppermann, M.Specht, A Context-Sensitive Nomadic Exhibition Guide, Proceedings Symposium on Handheld and Ubiquitous Computing, pp.127-142, LNCS 1927, 2000.

[6] F.Paternò, C.Santoro, One Model, Many Interfaces, Proceedings CADUI 2002. Kluwer Publisher.