UNIVERSITÀ DI PISA

Scuola di Dottorato in Ingegneria "Leonardo da Vinci"

Corso di Dottorato di Ricerca in
INGEGNERIA DELL'INFORMAZIONE

Tesi di Dottorato di Ricerca

USER INTERFACES FOR MOBILE DEVICES: TECHNIQUES AND CASE STUDIES

Autore:

Giuseppe Ghiani

Relatori:

Prof. Fabio Paternò

Prof. Luca Simoncini

Anno 2012

ABSTRACT

The interactive capabilities of portable devices that are nowadays increasingly available, enable mobile computing in diverse contexts. However, in order to fully exploit the potentialities of such technologies and to let end users benefit from them, effective and usable techniques are still needed.

In general, differences in capabilities, such as computational power and interaction resources, lead to an heterogeneity that is sometimes positively referred to as device diversity but also, negatively, as device fragmentation. When designing applications for mobile devices, besides general rules and principles of usability, developers cope with further constraints. Restricted capabilities, due to display size, input modality and computational power, imply important design and implementation choices in order to guarantee usability. In addition, when the application is likely to be used by subjects affected by some impairment, the system has also to comply with accessibility requirements.

The aim of this dissertation is to propose and discuss examples of such techniques, aimed to support user interfaces on mobile devices, by tackling design, development and evaluation of specific solutions for portable terminals as well as for enabling interoperability across diverse devices (including desktops, handhelds, smartphones).

Usefulness and usability aspects are taken into great consideration by the main research questions that drove the activities of the study. With respect the such questions, the three central chapters of the dissertation are respectively aimed at evaluating: hardware/software solutions for edutainment and accessibility in mobile museum guides, visualization strategies for mobile users visiting smart environments, and techniques for user interface migration across diverse devices in multi-user contexts.

Motivations, design, implementation and evaluation about a number of solutions aimed to support several dimensions of user interfaces for mobile devices are widely discussed throughout the dissertation, and some findings are drawn.

Each one of the prototypes described in the following chapters has been entirely developed within the research activities of the laboratory where the author performed his PhD. Most activities were related to tasks of international research projects and the organization of this dissertation reflects their evolution chronology.

CONTENTS

1. INTRODUCTION

1.1 Mobile computing

Thanks to the technological advances, various types of portable devices with increasing capabilities are nowadays available to the great public. A large set of functionalities is included in modern interactive devices, and makes mobile computing really possible in different contexts. However, although the potentialities of such technologies are promising, lack of techniques to fully exploit them often hinder to bring their benefits to everyday users.
When designing applications for portable devices, besides the universal rules and principles of usability, developers have to cope with additional constraints. Indeed, due to the typically limited resources of portable devices with respect to desktop/laptops (e.g. small-sized display, touchscreen instead of keyboard or mouse, etc.) important design and implementation choices are required to developers in order to guarantee application usability. This is even more true when applications are likely to be used by subjects affected by impairment (e.g. blindness), as in this case the system has to be accessible.
Differences in capabilities, such as computational power and interaction resources, lead to an heterogeneity that is sometimes positively referred to as device diversity but also, negatively, as device fragmentation.
The case studies and techniques discussed in this dissertation, tackle both the design and development of specific hardware/software solutions for portable devices (e.g. mobile guides for indoor exhibitions) and the interoperability of applications across diverse devices, including mobile ones.

1.2 Dissertation motivations and objectives

This dissertation summarises the results of a research work that has been carried out at the HIIS (Human Interfaces in Information Systems) Laboratory of CNR-ISTI in Pisa, Italy. Most activities were related to tasks of international research projects. The subsequent chapters discuss motivations, design, implementation and evaluation about a number of solutions to support several dimensions of user interfaces for mobile devices. The main research questions that drove the creation of the various solutions discussed in the following are about usefulness, usability and accessibility aspects.
The aim of the first investigation, that copes with the case study of a *mobile museum guide*, is to evaluate how edutainment and accessibility can be supported by the use of mobile guides in real museum environments. In detail, an enhanced solution combining indoor RFID localization, educational games and multi-device interaction is compared to a traditional mobile guide, with the aim to study user's attitude towards such novel supports. A first research question is whether and to what extent enhanced mobile solutions are actually able to foster learning among visitors of exhibitions. An investigation extension is related to the accessibility aspect of the location-aware mobile guide equipped with *vibrotactile feedback*. In particular, a further question is whether the multimodal mobile guide is a valuable support for visually impaired users visiting an exhibition in autonomy.
The second investigation focuses on the extension and application of localization supports and visualization techniques, developed for the museum case study, on a

different domain, i.e. *smart environments*. The objective is to evaluate the usability of such techniques for target device selection, in particular for migration of the user interface from mobile to stationary devices. The research question is about which visualization modality of the smart environment map is more intuitive for the mobile user when selecting a target device for migration.
The last investigation deals with migration of user interfaces, and discusses various dimensions of a *Migration Platform*, from which design and development several research questions iteratively arise. Examples of such questions include the usability of the strategies for user-driven component selection for partial migration from desktop to mobile devices, the impact of privacy concerns when multiple users are involved in migration of the same interfaces, and the security level of the overall Migration Platform.
In order to answer to the specific questions, each proposed investigation reports on one or more studies based on usability tests.

1.3 Dissertation structure

The organization of this dissertation reflects chronologically the activities carried out by the author during his PhD period.
The following chapter describes a twofold case study of an educational and accessible mobile guide for museum environments. Edutainment, multi-device, multimodality (vocal and vibrotactile output), location-awareness aspects are tackled with respect to a solution that has been developed for a real museum environment.
A study on representation of smart environments in mobile devices is reported in Chapter 3, and shows how some of the techniques that were initially designed for accessibility purposes (i.e. for impaired individuals) can be reused in different domains, such as in selection tools for target devices for interface migration.
Chapter 4 is devoted to discuss the iterative evolution of a Migration Platform, tackling aspects such as multi-user application access and privacy/security, also reporting on a series of usability and technical evaluations.
Each of the central chapters on the dissertation, besides discussing motivations of the work and technical details of the developed prototypes, also summarizes the findings and mentions the open challenges. Each one of the described prototypes has been entirely developed within the research activities of the laboratory where the author performed his PhD.
The last chapter reports on the conclusions that can be drawn the dissertation and provides possible further developments.

2. USER INTERFACES FOR MOBILE GUIDES: A CASE STUDY

2.1 Introduction

Thanks to the latest technological improvements, novel interactive software environments are able support users in different domains and with different aims.
Museums are a particularly interesting domain in which to investigate new interaction modalities to guide visitors and to improve their learning experience.
The investigation reported on this chapter has started by considering a previously existing version of a mobile guide for museum environments [31]. Due to the technology improvements of the last years, a new prototype has been designed from scratch by exploiting richer programming frameworks and finer location sensing techniques. The various versions that have been implemented so far, have made possible to investigate combinations various capabilities, such as location awareness, multi-device interaction and multimodal output.
This chapter tackles two dimensions of user interfaces for museum mobile guides: edutainment and accessibility. Although there is partial overlapping, such aspects are tackled in two separate sections by describing the development of two slightly different solutions and their evaluation. The user studies that have been carried out have led to interesting results, which not only are useful with respect to possible improvements for the implemented prototypes, but also for being reused in future research work.

2.2 Edutainment

Those forms of entertainment with educational purposes are often referred to as edutainment. Examples of media with teaching aims may include television shows, Web sites or applications in general.
Due to the large amount of digital information available and the increasingly technological resources adopted, museums can be considered as a concrete example domain for edutainment. Indeed, the diffusion of the Web and content digitalisation allow to overcome physical barriers. Museum collections can be accessed even by virtual visitors, who can virtually visit the museum while searching for items and thus get an experience as much similar to that of the real ones.
The huge amount of digital data already available in these domains makes museums a particularly suitable context in which to study how new interaction modalities can guide the visitors. On the one hand, such novel techniques can improve the mobile users experience. On the other hand, if not adequately supported, the wealth of both available information and devices in use might become a potential source of disorientation for users.
Support for museum visits has traditionally been limited to portable audio guides and interactive kiosks, which suffer from various drawbacks (such as the impossibility to run general purpose applications). It is thus reasonably important to exploit new technologies for identifying new solutions also able to promote social interaction [64] while enhancing user experience. Many studies, such as [56] and [65], have underlined that the interaction with exhibitions, as well as communication and interaction among visitors are important for a successful learning environment.

The number of visitors that museum settings can involve at the same time can be exploited to improve user experience, with respect and edutainment/infotainment (i.e. learning) aspects, both at an individual and a cooperative level. In addition, if the mobile guides provide some forms of games (individual and/or collaborative) built on the content of the visited museum, the visitors would be encouraged to challenge themselves and other visitors. This competitiveness provides an interesting and amusing way to promote user interaction and learning in such settings.
In this regard, the use of different types of devices within the museum (e.g., mobile devices and large screens) can be a way to enrich user experience by enabling enhanced functionalities. Showing the visitor's position and presenting individual and cooperative games on the large screen while the mobile guide supports the user in the museum visit are examples of such features.
User interfaces intelligence, including provision of adaptivity and personalisation, is well suited for augmenting educational museum visits. Indeed, it is worth considering that understanding and learning is facilitated when the systems rely on concepts familiar to the users (with respect to their interests and knowledge level). Automatically adaptation of the content presentation by exploiting data stored in a user profile is an example of strategy for supporting and facilitating user learning in this regard.

Recent increases in the performance of PDAs (Personal Digital Assistants) and mobile phones have led to a growing interest in the development of mobile guides. The progress of wireless communications and localisation technologies offers the possibility, by following the context-aware computing paradigm, to provide users with context-dependent services, exploiting information such as user's location, nearby people/devices, and current task.
One of the first examples of mobile guides with context-awareness was Cyberguide [1], which supported navigation inside buildings as well as outdoors. The visualization of a schematic map of the visited area was automatically updated according to the user's position. The hardware enabling location awareness consisted on a set of infrared sensors deployed in the environment (indoors) and on a GPS receiver (outdoors).
Research work has also investigated the dynamic generation and delivery of personalised comments to the user. An example of adaptive mobile guide is HyperAudio [87], that generates audio comments for a palmtop according to the physical location of the users. The amount of time spent in a certain location is considered as an indication of user interest in the related (nearby) artwork(s). HyperAudio has recently been considered by some of its own authors, in order to analyse whether the results that were gathered at that time are still valid in light of more recent experiences and findings [86].
An intelligent context-aware tourist guide for city visitors was developed within the GUIDE project [29]. Information on user position is used to provide an automatic information delivery service: each point of interest is associated with a geographic area, each one covered by a Wireless LAN. When the user enters an area, the related information is automatically furnished as text descriptions, images and audio comments. In GUIDE, the notion of context comprises not only to users' location but also their personal profile as well as external conditions, such as opening times of city attractions.

Hippie [83] is another example of mobile guide exploiting several types of information to adapt its services. User location is merged with other information to provide additional details on the exhibits. Hippie furnishes comments on the artworks in sight. Information is adapted to user's location, but also to his/her interests and knowledge derived from prior interaction. However, automatic generation of comments about the closest artwork may sometimes be judged annoying by the users.
CRUMPET [89] personalises also its services, available both for PDAs and mobile phones, according to the position of the users, their interests and previous interaction with the system.
Lol@ [90] is a tourist guide for the city of Vienna, using GPS for localisation and GIS support for map generation. Information is adapted to the device in use, but not to the user's characteristics. VeGame [13] uses mobile technologies to explore the city of Venice and learn about history and architecture through (video) games based on observation, reflection and action. Although bandwidth is limited, the system supports wireless communication between peer PDAs. Visitors may play in teams, with the goal to achieve the best score. Each team can have multiple members but only one PDA.
The above mentioned approaches to mobile guides have brought useful and important contributions, but none of them has explored the possibility to use large screens that can dynamically be available for the visitor.
The City project [22] was carried out as part of the Equator project at the Lighthouse Museum (Glasgow, UK). The proposed system is built upon several technologies. For the real visit, a PDA equipped with headphones, a microphone and an ultrasonic location system are used. For the virtual reality visit, a 3D representation of the museum is navigated. For the Web visit, a standard browser with Java applets is used. By relying on this system, visitors can share their museum experience and navigate in cooperation on mixed contexts: the Web, the virtual and physical reality. Location and orientation of each visitor are shared among users, that can also communicate through audio channels. The authors have observed that voice interaction, location/orientation awareness and mutual visibility are essential for a successful museum co-visit involving remote users. While the Equator City project combines the real and a virtual representation of the museum, the work presented in this section is instead aimed at supporting ‘‘physical” visitors moving through a real museum/exhibition.

There is increasing interest in the use of RadioFrequency Identification (RFID) technology in order to enrich physical artworks with digital information. An examples can be found in [71], that proposed the scan and tilt interaction paradigm: physical selection is done by scanning RFID tags on the artworks, and single handed tilt gestures control and navigate the user interface and multimedia information. By pointing at the artwork of interest and controlling audio information with hand gestures, overloading of the visual channel is avoided, resulting in a less intrusive interaction technique. A first empirical evaluation of this prototype was performed, and the user study revealed a good acceptance among participants. However, at the same time, the investigation highlighted some limitations: the passive RFID tags used in the prototype forced the users to stand in very close proximity to the artworks, which is not realistic and uncomfortable in museum environments.

The mobile guide described on [12] was specifically designed to support blind people while visiting an exhibition. Users were provided with multimedia descriptions about points of interest in an event-driven manner: when a user entered an area related to an RFID tag the software proposed the corresponding description. However, some zones of the exhibition were not covered by any tag.
One of the benefits of the solution proposed in this section is to overcome the limitations of the two above mentioned works. For example, according to the proposed solution, the entire area of the environment is coverable by RFID tags, in order to provide the user with his/her position inside a section and to monitor user movements. Data on the path covered by the user are continuously compared with the artwork-related information explicitly accessed through the PDA interface. Such strategy of user modelling has an adaptation purpose that will be explained in later subsections.

RFID technology has also been exploited to improve the social interaction between users by the History Hunt project [46], implemented in the Nottingham castle museum. Groups of visitors (usually families), on arrival at the museum, collect a set of clues written on pieces of paper electronically tagged with RFID tags leading them in search of a given historical figure. Visitors interact with displays installed around the museum and receive further information about the clue.
By means of this application, collaboration occurs only when visitors are in close proximity to the interactive displays. The approach presented in this dissertation, supports instead collaboration throughout the whole visit (more details are provided in the following subsections).

The use of games to enhance interactivity in museums has been considered promising, as visitors would have the opportunity of improving their learning experience by challenging themselves and/or competing with others. While the games have been extensively explored through conventional media, such as paper-based treasure hunts, little research has been reported on the role of digital adaptive games inside galleries. Similar research [11] has been carried out in other contexts.
The development of a museum guide prototype for the Canadian Museum of Nature in Ottawa is presented in [53]. The guide is an augmented audio reality system with location-aware capabilities based on RFID and optical position tracking. The authors have found that an accurate dynamic user model and recommender system can be built by exploiting ontologies and user interaction analysis.

A wearable museum guide that gathers visitor’s interests from the physical path covered is described in [101]. The user model is iteratively refined, in order to deliver personalised multimedia descriptions and to enrich the museum visit.
The PEACH (Personal Experience with Active Cultural Heritage) project [102] proposed a multi-device support to improve museum visitors experience with mobile and stationary devices. A virtual presentation agent transfers from the handheld device to the Virtual Windows (large displays) when such devices are in sight, and assists the visitor. Adaptivity is achieved by considering the elements that were likely to be of major interest to the visitor and by tailoring multimedia presentations. Among the claimed principles, the unobtrusiveness of the visit

support was the more relevant: the system should never intrude between the exhibit and the visitors, letting them focus their attention on the physical environment.
Differently from previous work, this chapter of the dissertation not only tackles the multi-device museum environments, but also novel solutions to exploit interactivity, for instance dynamic migration of part of the game interfaces (including social games) from one device to another.
The aim is to provide edutainment opportunities to visitors, to improve their individual experience as well as to stimulate socialisation among them.
Detailed discussions of such features are provided in the following subsections.

2.2.1. Overview of the mobile guide

The interactive environment for museum visitors discussed in this section has been applied to the guides of two museums: the Marble Museum (Carrara, Italy) and the Museum of Natural History (Calci, Italy). The Cicero digital museum guide [31] has been the starting point from which to develop the multi-device mobile guide. Cicero was developed for a PDA platform and freely available to the visitors of the Marble Museum. Such experience has led to the understanding of specific requirements of mobile museum guide users. First, users are not willing to spend too much time learning how to use it. In addition, they rarely visit the same museum more than once, and thus the mobile guide should be intuitive and easy to use. The mobile application should actually serve as a support/integration for the real museum experience, and therefore it should not be too intrusive or preclude users' appreciation of the real artworks. In order to fulfil such requirements for the mobile guide to be a useful learning opportunity for visitors, a number of features have been included in the prototype.
Users are provided with a rich variety of multimedia information about the available artworks. Information is mainly provided vocally, in order to allow visitors to freely look around, but graphical resources are also available. Indeed, the visual interface can show videos, maps at different levels (museum, sections, rooms), and specific descriptions. The guide also supports location-aware services such as showing the path to a specific artwork from the current location. The location-aware features of the mobile guide are discussed in the following subsection. Personalisation/adaptation capabilities have been included (and are also described in the following) in order to provide information that is considered interesting for the user.
Several games are available as a form of learning activity, both as single-user and multiple user games (see subsection 2.2.1.3), in order to improve user involvement and interaction. Lastly, subsection 2.2.1.4 is entirely dedicated to the description of the multi-device capabilities and subsection 2.2.1.5 provides an overview of the general system architecture.

2.2.1.1 Location-awareness

The mobile guide supports location-awareness as a means for improving its flexibility and efficacy. By providing users with context-dependent information and

services, a more appropriate support can be offered because it is possible to infer what the most relevant information is and deliver it.
The proposed location-aware mobile guide lets the user navigate both a physical space and a related digital information space at the same time, thereby offering a more complete experience.

Location-awareness is enabled in the mobile guide by long range active RFID technology, which allows a less intrusive localisation approach. Active tags can be detected even if the reader is not be in very close proximity, which is a useful feature in real museum settings..
Figure 1 shows the implemented museum guide on the PDA device equipped with the Compact Flash RFID reader, which is a small add-on that can be plugged into a PDA.

Figure 1. The Museum Guide application in a Mobile Device with RFID Reader.

In the proposed system museum artworks are fitted with RFID tags, each one having a unique ID. It has been considered reasonable to associate a single tag with more than one nearby artwork, when they are very close. This is due to the difficulty of distinguishing two or more tags that are very close to each other. Indeed, if two tags were placed in a very small area, the reader would detect both of them with the same RSSI (Received Signal Strength Indication). Such problems related to tag density, and the use of RFID technology for localisation are also discussed in [12].

As soon as the user enters a new room, a short report summarizing the visit of the previous section (e.g., the list of visited artworks) is generated. Then, the guide displays the museum map highlighting the room being visited and plays the vocal description of the room. When the user is close to an artwork, the guide vocally asks whether s/he needs additional information; given an affirmative response, the information is provided vocally with further related graphical content (such as videos, if available).

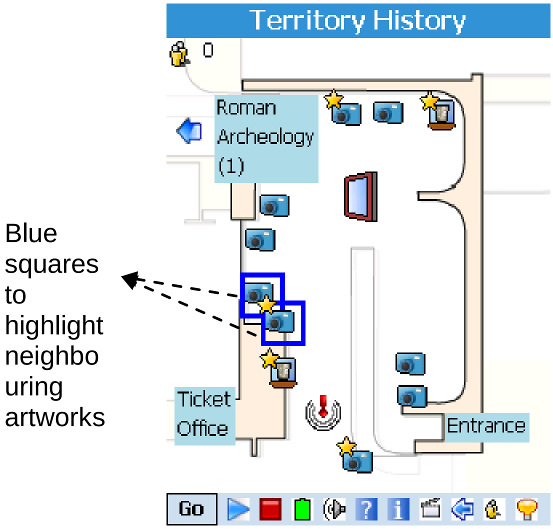

Figure 2: The artworks closest to the user are highlighted by blue squares.

Continuous monitoring of RFID tags enables the guide to calculate the artworks closest to the user. When a new tag is detected, i.e. a new area is being visited, an audio clip is played to alert the user, and a vocal message indicating the number of nearby artworks is generated. Depending on user preferences, the guide may ask or not confirmation before describing the detected artworks. Figure 2 shows how the artworks detected via their tags are highlighted through a blue frame around the corresponding icon in the map.

Figure 3 (left side and right side) shows how the guide dynamically updates map information as the user moves about. The three artworks displayed in a different colour on the left screenshot in Figure 3 represent those already visited. As the user accesses the descriptions of two different artworks (right side) the colour of the associated icons changes to show that the related artworks have been viewed.

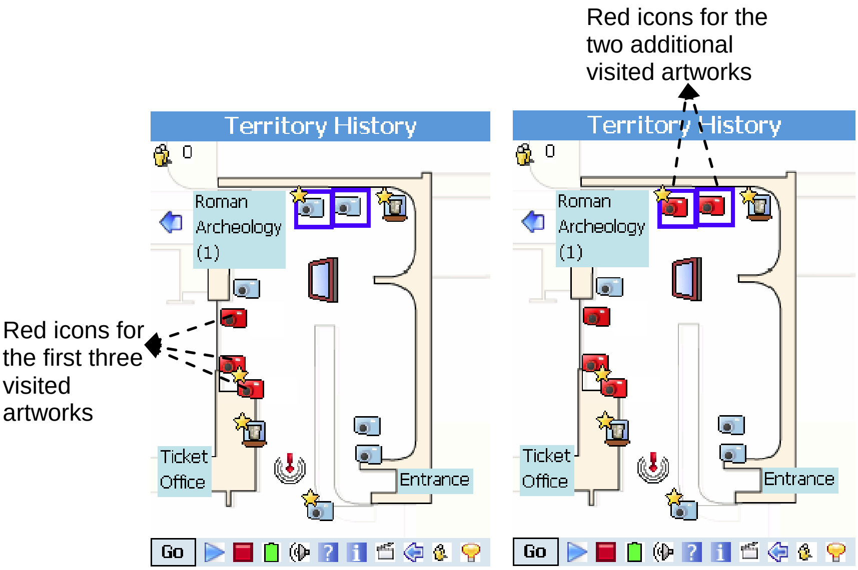

Figure 3: Dynamic update of location-aware map information.

2.2.1.2 Adaptive support

Adaptive features enable a degree of personalisation in the mobile guide user interface. The adaptive support is based on a user model which is designed to record information on user interests and preferences. The user model is dynamically updated according to events that occur during the visit. Indeed, the user movements are tracked. Information regarding user's level of interest towards the various museum items is inferred and enables the provision of personalised information to the user.

The application logs any interesting event, such as section and area changes (with respect to nearby artworks) and how much time is spent within such areas. It is then possible to recall the path covered through the museum. The time spent by the user listening to an artwork description is also logged. Artwork and room descriptions are vocally reproduced through a TTS (Text To Speech) engine. The main benefit of a dynamically generated audio description is the limited memory required for storing audio resources. Indeed, the audio files do not need to be physically saved on the PDA but they are dynamically created by the TTS starting from textual information. Additional museum information can thus be easily managed because the audio files have not to be manually created but are dynamically generated by the TTS. This is particularly significant when the descriptions have to be available in different languages. The Embedded TTS provided by Loquendo (http://www.loquendo.com) has been used as the vocal engine.

Logged data are used to keep the user model updated. By analysing it, the guide estimates in real time user favourite type of artwork and assesses his/her preferences. At each section change, the user model calculates which artwork in the new section would best match the user preferences. The suggested artwork is then highlighted by a light-bulb icon next to the corresponding artwork (see Figure

4). Details on how user's interest estimation is presented on the large screen device are provided in the following sections.

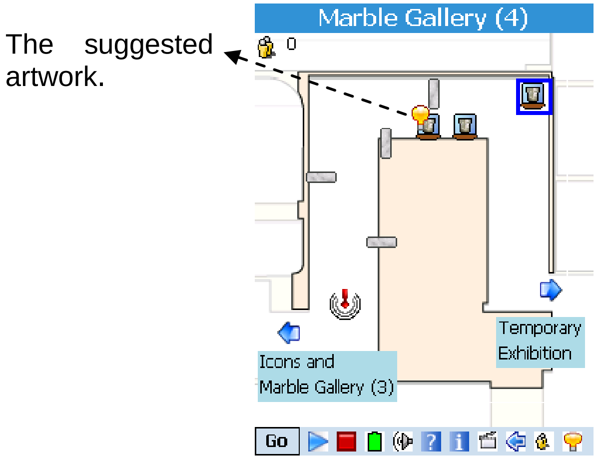

Figure 4: Highlighting the suggested artwork by a bulb icon.

User preferences are estimated by the application according to several sources of information. The data analysed to infer such preferences includes the physical and the virtual visit performed by the user (e.g. access to descriptions of artworks even without being in their proximity). It is worth pointing out that, since one of the requirements of the mobile guide is to be unobtrusive, no explicit user intervention is required to fill the user model: data are only derived from the visitor's behaviour.

Each access to a museum item performed through the mobile guide is interpreted as an indicator of interest/preference of the user towards the characteristics of that item. Author, period, material and artwork category (e.g., sculpture, painting, picture) are considered in the computation of the preferences. Also, the time spent on listening to the description of an artwork is interpreted as an implicit sign of interest and therefore affects the specific weight of the values of the various attributes associated with a particular artwork in the user model. In addition, playing any game also influences the user model: playing a game is considered as an indication of interest, which is further augmented if the user correctly solves it. In addition, if the user also physically lingers in front of the artwork (and consequently the application logs that s/he has entered the corresponding artwork's RFID tag area) then a higher level of interest is recorded with respect to that kind of artwork.
Local suggestions are also available on request: when the light-bulb button is selected in the toolbar, the artworks in the current area that have not yet been visited are considered as possible next candidates and a light-bulb icon besides an artwork indicates the most interesting artwork among the nearest ones.
The user model also exploits the possibility to generate audio descriptions on the fly. The insertion or deletion of some parts of the automatically read text is based on the estimated user preference, tailoring the length according to the calculated interest level.

The User Model module consists of two main parts, one maintains the data and the other one infers user preferences and updates the model. The former stores data about user, usage and environment; the latter reads logged data and updates the model exploiting predefined stereotypes and heuristics.
User model updating procedure starts when the user interface notifies logger about an interaction. The logger, in turn, generates the appropriate event. Each user model component that has subscribed to this event infers data with its own algorithm. When the presentation layer reads the status of the user model, it adapts the user interface accordingly.
Visitors usually go to museums in order to acquire knowledge about the proposed collections and/or to gain new interests. In addition, while visiting the museum, users tend to become more familiar with the collections and to progressively focus their attention on specific types of items. This is especially true when the exhibition is very large. For this reason, the proposed user model also adopts a recentness-weighted approach that assigns higher weights to the ratings of the most recently accessed artworks. Such ratings are used by the guide to provide the user with suggestions regarding the next artwork(s) to visit (in the user interface, such suggestions are rendered through light-bulb icons).
Whenever the user accesses a certain artwork, the algorithm updates the vector containing the current ratings. The calculation of the rating for the current artwork is performed by taking into account the ratings of the previously accessed artworks that are correlated with the current one. For the first access, the rating is calculated by considering predefined stereotypes to estimate level of interests for a certain topic/artwork. Each artwork in the museum is initially assigned a default rating value, which is statically calculated; afterwards, the values are calculated dynamically.

2.2.1.3 Educational games

Games related to the content of the real visit are an effective tool for improving visitors' educational experience and stimulating more active participation. By means of educational games, visitors can challenge themselves and others about the information received, in order to check what they have actually retained.
The latter point is not trivial as the information conveyed can be quite overwhelming, especially in large museums. Thus, games are in general a way to check the knowledge users should have actually gained. Individual games support learning at users' own pace, while collaborative games are aimed to stimulate talking and discussion about the subject of the game. In this context, the use of a large screen is meant to facilitate discussion among a large group of people.
Preliminary ideas on the use of games in mobile guides were introduced in previous work, though without reporting on any localisation [41] or on multi-user support [49]. The inclusion of games within such systems was considered as a good way to test what was learned during the visit and to make the museum visit more interactive.
The entire solution presented in this section has been tested into two museums, with a wider set of games and an associated game editor. Such games are integrated with the user model that enables some adaptive features.

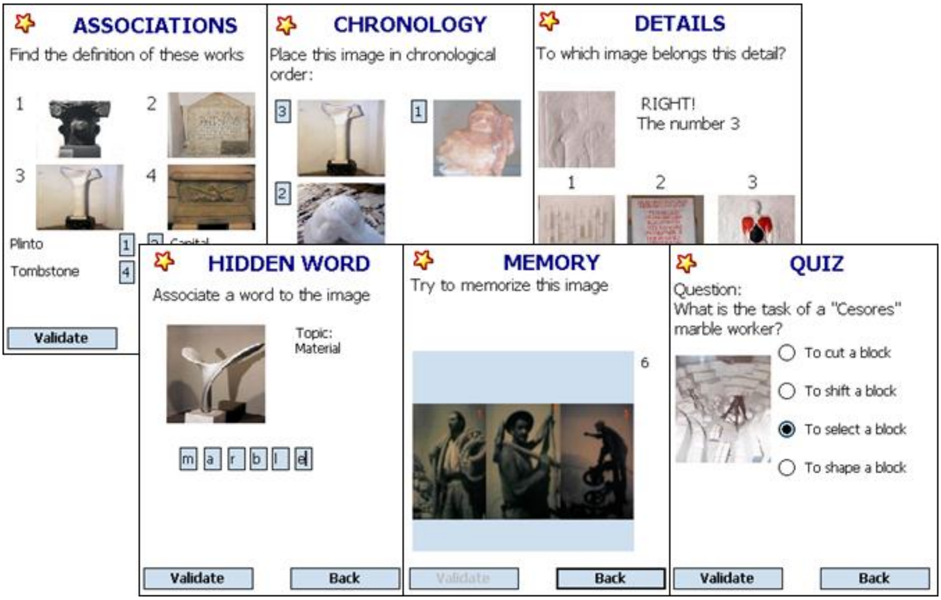

Figure 5: The six types of individual games.

<u>*2.2.1.3.1. Individual games*</u>

In order to increase the learning experience, six types of individual games related to the museum content were defined (see Figure 5):

- *Associations*. Images have to be linked to words (e.g., the image of the artwork to its author or material).
- *Details*. An enlargement of a small detail of an image is shown. The player has to guess which of the artwork images the detail belongs to.
- *Chronology*. The images of the artworks shown have to be sort chronologically, according to the artwork date.
- *Hidden word*. A word associated with a particular attribute of an artwork has to be guessed: the number of characters composing the word is displayed as help.
- *Memory*. An image is shown for a while, and then the user has to answer to a question about it.
- *Quiz*. A multiple-choice question with a unique correct answer.

Playing interaction is done by simple clicks or text insertion through radio buttons or text boxes. Thus, games may be solved in a few seconds and are suitable to visitors of a wide age range.

As depicted in Figure 6, the icon of the artworks with game associated is enhanced with an additional star-shaped symbol. The game is accessed by clicking on the symbol.

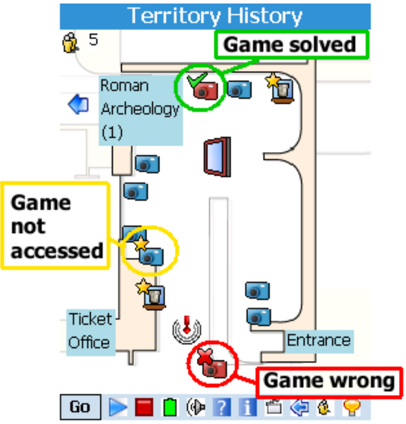

Figure 6: Representations of the game state.

<u>*2.2.1.3.2. Individual games*</u>

Social interaction is acknowledged as a relevant strategy for improving learning, as discussed in [56, 65, 50, 33]. Group games, in which users are organised in teams, have been included in the museum guide in order to stimulate cooperation among visitors whilst improving social interaction. The *shared enigma*, introduced in [31], is an example of group game: a series of questions on a topic are associated with an image hidden by a jigsaw puzzle. Each player must solve an individual game to reveal one piece in the puzzle, which is shared by the players of the same team, thus helping the team to answer the question of the puzzle. As soon as an individual game is solved, the group earns points.

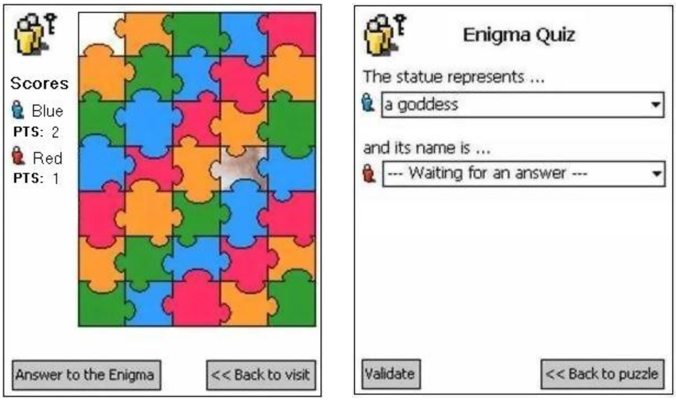

Figure 7. The shared enigma on the PDA.

Figure 7 shows the PDA interface of the shared enigma: the first part of the interface displays the current players' scores and the hidden puzzle image, the second one shows the questions and the possible answers.

Figure 8 shows how the various players in the same team can have mutual awareness of each other: coloured bullets are placed beside the presentation of each visited artwork Each bullet is related to a different player in the same team, so that a player can see which artworks have already been accessed by the others. Differently from Figure 8, Figure 7 shows how much score each player (identified by a colour) has gained up to the current moment.
The current games, though quite simple, received positive feedback by the users involved in the evaluation sessions. More details are reported in the subsequent sections.

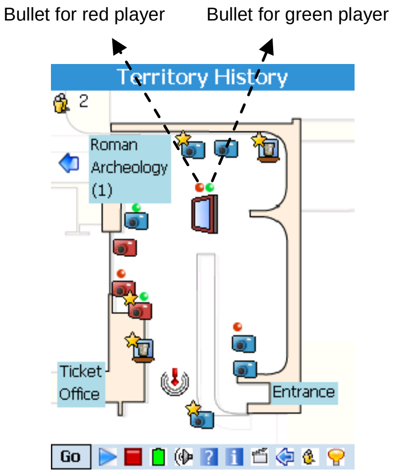

Figure 8. PDA Interface for the Cooperative Version.

2.2.1.4 Multi-device support

Museum visit and games are supported by exploiting both mobile and stationary devices equipped with large screens. The typical scenario is users interacting through the mobile device, while freely moving around the exhibition environment, who can also exploit a shared large screen of a stationary device when it is nearby. [21] dealt about the ‘‘honey pot effect” arising from the interaction with public situated displays: shared screens connected to stationary systems tend to promote social interaction and to improve user experience, otherwise limited to individual interaction with a mobile device. Public displays also stimulate social interaction and communication with other visitors, even when they do not know each other.
In the museum guide domain, a larger shared screen extends the functionality of the mobile application: individual games are presented differently, social game representations may be shared, the positions of other players within the museum map are shown, and a virtual pre-visit of the entire museum can be performed.

Shared displays can be in two main states: *standalone* or *split*. In standalone mode the screen has its own input devices (keyboard and mouse) and can be used for a virtual visit of the museum by visitors who do not have the PDA. When a display is in split mode, it means that one visitor has taken control of it through the PDA.

The communication between mobile device and large screen is described later on. Shared displays support several types of accesses, thus the layout and some parts of the interface remain unchanged, being similar to the PDA interface in order to avoid disorienting users. The permanent part of the user interface provides information such as the map of the currently visited section, its location within the museum, an explanation of the icons used and the state of the shared enigma. In standalone mode users can select from three types of views of the section map: *icons*, *thumbnails* and *thumbnails plus icons*. Icons represent the artwork type while thumbnails are small photos of the artwork.
The large screen application exploits user model data from the connected PDA and generates an ‘‘interest” rating for each artwork of the selected room. Ratings, which are expressed by ‘‘LED light bars” on a scale of 0–5, thus providing a more detailed indication with respect to the suggestion shown in the PDA.

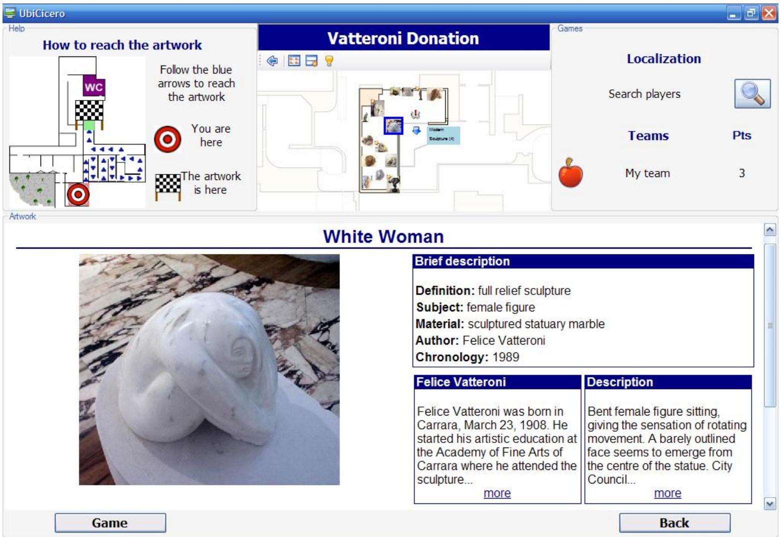

Figure 9: Artwork presentation on large screen.

The large screen, besides what is displayed on the PDA, shows additional information about the artwork, such as the shortest path to get to it (see Figure 9). A high resolution image, annotated with pieces of information that do not fit in a single PDA presentation, are also shown in a single presentation.
Whenever a player selects the connection through the PDA interface, the large screen changes its state into split mode. In this context the large screen is used both to show additional information and to focus the attention of multiple users on a given game, relying on the screen size. When a player is connected to the large screen, the section map view automatically switches to thumbnails mode and the artworks type is shown on the PDA (see Figure 12). The artwork presentation relies upon a higher resolution image on the large display in order to add more description information.

Even the visualisation strategy of the shared enigma depends on the availability of the large screen. If only the PDA is used, then the interface consists of two presentations sequentially displayed on the PDA: the hidden image and the associated questions. If the larger shared screen is available, then the hidden image is shown on the large display, while the questions and the possible answers are presented on the PDA.
Providing an effective representation of players' position on the PDA is rather difficult due to the small screen area available. This is especially true when players are located in different rooms. The large shared screen can instead be used for this purpose: it is divided into sections, one for each player. Each section shows the name and the room where the player is located and the artworks close to the player are highlighted by rectangles (see Figure 10). The location of the player is automatically obtained through the detection of the RFID tag(s) near each visitor (this information is forwarded wirelessly by the PDAs to the computer controlling the large screen).

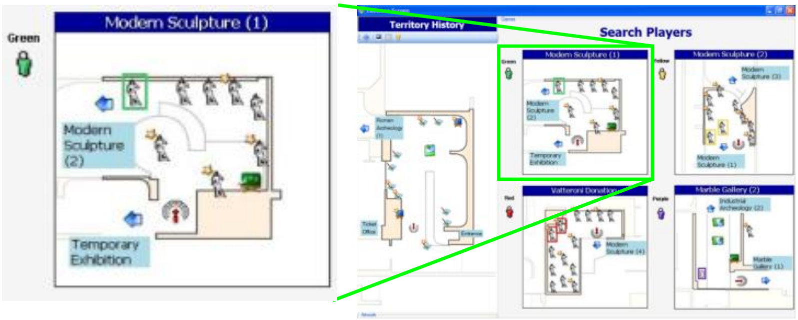

Figure 10: Showing players' positions with respect to the visited artworks on large screen.

2.2.1.5 Software architecture
The software architecture of the multi-device and location-aware museum guide is composed of several modules allocated on different devices. The main modules of the software architecture are deployed in the PDA and in the stationary device. The communication protocol of the environment is also part of the architecture, since it has been specifically designed for this purpose.
The PDA module is composed of five layers, each of them provides the others with services (see Figure 11).

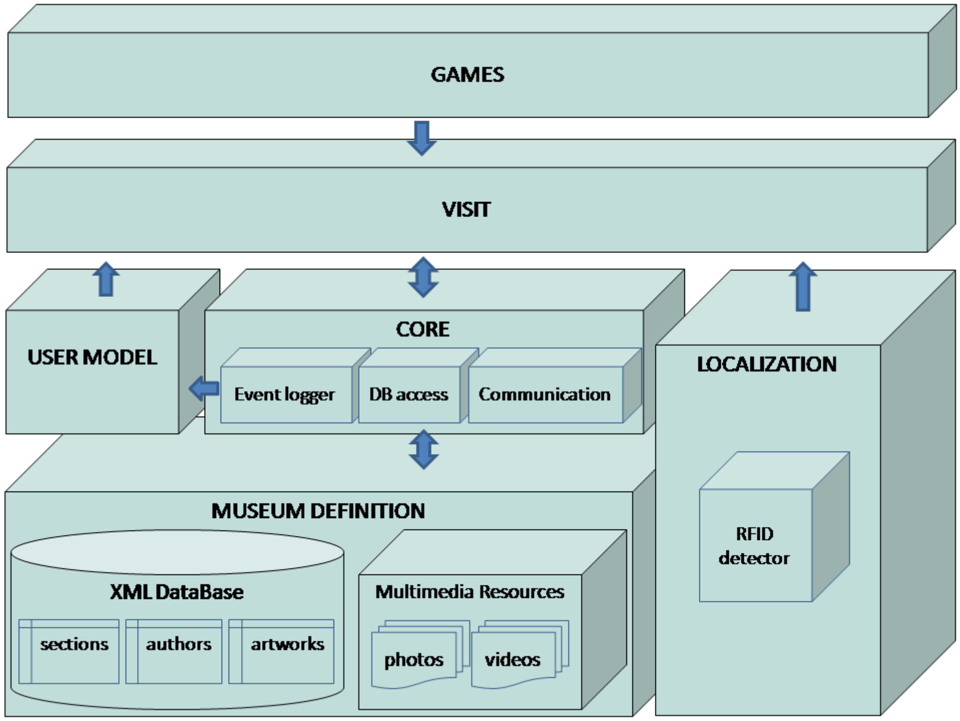

Figure 11: The software architecture of the mobile guide.

From a bottom-up perspective, the modules are summarised in the following.

- *Museum definition*, containing the specifications of the whole museum:
  - XML database.
    - Sections' layout (room sizes and shapes, and item positions).
    - Description of sections, artworks and authors.
  - Multimedia resources such as photos and videos concerning the exhibited items.
- *Localisation*, enabled by the RFID detection function.
- *Core*, defining routines for database access from the upper layers, for communication with other devices and logging.
- *User model*, which contains the level of preferences regarding various information attributes.
- *Visit*, supporting interactive presentations for accessing museum information.
- *Games*, providing edutainment features.

The architectural blocks above mentioned are shown in Figure 11. It is worth pointing out that not all the functionalities must necessarily be present at run time, since several modules are detachable from the application (both for stationary and for mobile devices) by means of conditional compilation.

Therefore, if a certain museum environment is not going to adopt a RFID localisation infrastructure, then the application version for that particular environment would be re-compiled omitting the RFID detector, and this will be done without the need of restructuring the whole source code. Further details regarding such modules are provided in the following subsections.

2.2.1.5.1. Museum definition

The museum environment is defined by an XML database and a set of multimedia resources. The database specifies shape, size and layout for each section of the exhibition and also stores the artwork-RFID tag correspondences. Such an association is fundamental to automatically highlight the currently visited item(s) at run time (see Figure 2). The multimedia files add information to sections, artworks and authors that the user may want to look at. Photos and videos showing the creation of a contemporary sculpture are an example of such resources.

2.2.1.5.2. Localization

User position is shown on the PDA by highlighting the icons of the artwork(s) nearby. Blue frames are displayed around the corresponding icon(s). Localization is enabled by RFID technology.

RFID-based solutions are composed of two main parts: the tags and the reader. Tags basically have a static identification number (ID). The reader scans for available tags, and reports on their ID.

RFID technology can be applied by using *passive* or *active* tags.

Passive RFID tags do not have any internal battery and exploit the energy electromagnetically inducted by the reader antenna. Thus, passive tags can respond with their IDs when and only when a reader within a few centimetres queries them. Such devices can be extremely small and inexpensive. Passive tags on the 13.56 MHz frequency were used in a previous version of the discussed museum guide application [71].

Active RFID tags are equipped with an internal power source and can thus autonomously transmit Radio Frequency (RF) signals at any time. There are two different types of active tags, depending on the way they work: *beacon/broadcast* and *respond*.

Beacon/broadcast tags have a radio transmitter and continuously send their data at certain intervals of time. The reader tunes into the proper radio frequency and listens to the tag information. Beacon technology uses read-only tags.

Respond tags (which, in contrast to the previous ones, are read/write tags) wait for a reader request before responding. Due to the need for both transmitter and receiver modules, respond tags involve more complex architectures, larger dimensions and higher costs than beacon tags. However, while the battery life of beacon tags (2–4 years) depends on how frequently they transmit, the battery life of respond tags depends on how often they are interrogated. Generally, the typical lifetime of a respond tag battery is more than 6 years.

Active beacon/broadcast tags with 868MHz European frequency have been chosen for the guide application. The frequency choice was driven by law constraints (the experiments were performed in Italy). The technology choice was motivated by the capability of active tags to cover wider areas than the passive

ones. The adopted reader triggers tag detection at about 5m distance. The beacon mode has been used because it is more scalable than the respond solution (which experiences performance decay when the number of readers increases). The main benefit of the beacon mode is that the detection is not affected by the number of users, since the readers do not send any request to the tags (and therefore request collisions do not need to be handled). Actually, the reader just tunes into the proper radio frequency and ‘‘listens” to the tag(s) information, reporting the list of visible tags together with their RSSI (Received Signal Strength Indication). However, when multiple tags are sending their IDs, overlap may occur and data may be lost. For this reason, the detection of all the visible tags may take a few seconds (depending on the number of tags in range). To make the mobile guide as small and as light as possible, a totally handheld-based solution was tuned, consisting of Compact Flash (CF) RFID reader with small-sized antenna. The PDA does not need any additional expansion or adapter because the reader plugs directly into the embedded CF slot.
Interfacing to the RFID hardware is achieved via the libraries provided by the hardware supplier (http://www.identecsolutions.com/) whose functions allow the application to configure and interrogate the RFID reader. For each query (queries are performed with the frequency of two per second) from the localisation support, the RFID reader provides the list of visible tags. Every list element contains data related to a visible tag, such as ID and RSSI.
The localisation support maintains a list of all the tags that have been detected at least once. Each tag is associated with its last reported RSSI and the time elapsed since the last detection. Only tags that have been detected recently are considered in the computation. The best tag is always the one with the highest RSSI. However, a ‘‘new tag event” is generated only if the new tag is the best candidate for *n* consecutive queries. The value of n is specified in the application settings in order to achieve a trade-off between reliability and speed of the localisation: the higher the value, the more reliable the localisation will be, but also the more time it will take the identification of the closest artworks.
Therefore, the value of n must be chosen carefully, especially when tag density is high (i.e. artworks are very close), in order to avoid erroneous detections. To facilitate localisation an RSSI threshold is used to adjust the reader sensitivity. Lower sensitivity makes the reader report only the nearest tags and simplifies the computation.

2.2.1.5.3. Core

The core implements data structures needed by the upper layers (e.g., support for configuration and help) and the XML parsers. It also provides functionality to update the state of the players, to connect to shared stationary displays and to exchange information among PDAs, and therefore implements procedures for managing sockets, messages, and group organisation.
The Core also contains a set of methods for interfacing with the speech synthesizer engine, which is a package provided by a third part as it has been previously mentioned. The Core has three parts: the logging facility, the database interface and the communication module.

Event Logger manages the recording of relevant actions (e.g., location update, description access) on a log file, and raises events that are caught from the User Model to update the user profile.

DB access interface module provides a suite of functions to access the information stored in the Museum Definition layer. XML parsers and methods for getting the resources are defined in this module.

Communication functions are responsible for real-time information exchange among the devices in use. Indeed, since the state of the application is shared across the multiple active devices, real-time communication is fundamental for keeping the distributed state up-to-date.
As soon as a new device joins the session, it multicasts a discovery message. The first answering device (mobile or stationary) provides the new one with the current global state.
Communication may even involve two specific entities such as a mobile device and one of the stationary devices of the environment. The mobile explicitly requests connection to the large screen, and data exchange is possible since the former device is aware of the latter's IP address after the discovery phase. When the connection is established, the mobile device sends the shared state to the stationary device, together with the state of the mobile application (currently visited section, previously accessed artworks, solved games). Each event is then sent by means of update messages (score changed, artwork accessed, ...) to the large screen, which keeps its internal state up-to-date. The uploading capability is also fundamental for the stationary device to furnish a real-time feedback when the user is previewing the museum sections through the large screen.

2.2.1.5.4. User model

The user model defines the level of user preference regarding the various entities of the museum database.
Information regarding user movements, visited artworks and games result are used to continuously update the user model. Accessing a game associated with an artwork is considered as an expression of interest towards the corresponding knowledge, and, if the game is correctly solved, the level of interest is considered even higher. Likewise, when the user moves towards an artwork, then s/he is considered to be interested to it.

2.2.1.5.5. Visit

The Visit layer manages the presentation of a set of interactive elements, such as the current room map. Each artwork of the room is displayed by an icon that identifies its type (sculpture, painting, picture, ...) and that is positioned in the map according to the physical location of the real artwork.
By clicking on the icon, users get detailed information about the corresponding artwork. During the visit, it is also possible to access games, watch to videos, receive help, change audio parameters and obtain other info.
The guide application has been developed in C# over the Microsoft .NET Compact Framework. A custom graphical component, named *MuseumMap*, is embedded in the main form in order to allow the virtual the visit and to enable the interactive

features. MuseumMap subscribes to location update events produced by Localisation, queries the database for the current room layout (e.g., shape, type and coordinates of the items) and uploads the map representation highlighting the neighbouring artworks. MuseumMap also manages graphic-oriented computations such as scaling (for adapting the drawing area to the device display), and solves the correlation between the touched area and the referred object.

2.2.1.5.6. Games

Museum visit is extended by the games layer. Each game is defined through an XML-based representation according to its type. Modifications or additions are possible through any text editor. The game layer accesses the communication services, informs every player of the team about the score updating and invokes the XML parser.

2.2.2. Evaluation

The early informal user tests, that were initially performed, have been useful for providing basic suggestions about the user interface and for determining the most suitable type of RFID technology to adopt. After such early tests the RFID technology based on ‘‘Respond” technology was discarded, since its tags cover too large distances and are more difficult to manage.
Once obtained a more consolidated version, a first evaluation of the multi-device guide was carried out.
Seven users were involved in the Marble Museum and five in the Natural History Museum (8 males and 4 females), with an average age of 36.4 years old. Three users had secondary school education, the others were university graduates.
Users were requested to read a short introduction about the application, in order to learn objectives and main features. They were also instructed on the tasks to be carried out and were invited to test two versions of the guide. One version was equipped with the multi-device support, the RFID module for detecting proximity of artworks within a museum room, and the adaptive features triggered by the user model. The other version was a basic one, thus without multi-device support, adaptation capabilities or RFID support. The user interactions were automatically logged.
Users were asked to visit some sections of the museum using the two versions. In order to test the additional features provided by the enhanced prototype, they had also to solve a few games, and to perform at least one splitting (i.e. interface distribution) between PDA and a large screen within the museum. In order to reduce biasing, half of the users were requested to start the visit using the enhanced version and then continue with the basic one, while for the others the opposite sequence was followed. Afterwards, the users had to fill in a questionnaire.

Regarding the personal background, almost all participants reported not to have much familiarity with PDAs, even if some of them reported having used one for navigational applications or calendars. A few had previous experience in using digital museum guides, even fewer reported having used digital games in museum contexts.

Questions about technical aspects of the mobile guide were asked too, aiming at evaluating specific features (the vocal part, the games, splitting support, etc.), and at comparing the two versions of the guide. The quantitative questions asked for a rating on a 1-5 Likert scale, (with 1 as the worst and 5 as the best score).
Testers judged useful the position detection feature, i.e. surrounding the icon representing the artwork(s) closest to the user with a frame (M = 3.58; SD = 1.16). Their comments indicated that such information facilitates identifying artworks and associating them with an icon in the digital map, which is also useful for orientation goals, especially in large museums. Delays in the localisation support were noticed by some users, which caused some hesitations during the visit.
The support given by the enhanced guide for presenting descriptive information about the closest artworks was also judged useful (M = 3.75; SD = 0.87). However, some participants pointed out that additional interaction is needed for deactivating the description of uninteresting artworks.
The games offered in the PDA version were quite appreciated (M = 3.67; SD = 1.23) as well as in the desktop one (M = 3.82, SD = 1.47). The way in which the functionality was split between the PDA and the large screen was also rated quite positively (M = 3.2; SD = 1.48). With respect to the splitting, some participants declared that it might not be easy to follow a description that is displayed partially on a PDA and partially on a large screen, because of the consequent division of attention.
When they were asked whether they would have preferred to use keyboard and mouse or the mobile device for interacting with the large screen, six reported preferring the current (mobile) interaction support, four would have preferred mouse and keyboard, while two replied it was indifferent. The most frequent reason for preferring mouse and keyboard was the familiarity with such devices.
The ability to select the way to visualise artworks in a section, through icons and/or artwork previews on the large screen, received instead positive ratings (M = 3.38; SD = 1.5).
With respect to the multimodality, testers also reported whether they preferred the vocal part of the user interface or the graphical one. Six users preferred the vocal part, declaring that the vocal support allows looking at the artworks while listening their description, which is highly desirable in a context where the visual channel is often already overused. One user preferred the graphical version, three the vocal plus graphic version, two did not report any preference.
The effectiveness of the vocal part was rated positively (M = 3.67; SD = 1.22).
The majority of them (10 out of 12) preferred the enhanced prototype. Regarding the reasons given by the two who preferred the basic prototype version, one reported the ability to make a quicker visit, and the other felt that the basic version allowed for more user initiative than the enhanced version.
Benefits and problems were also reported about the enhanced prototype: on the one hand, some users appreciated the capability of providing more personalised information and the in-room localisation, which on the other hand was also reported as the cause of some delays in the application responsiveness.
The user interface of the games was rated relatively well (M = 3.67; SD = 1.37), though some users reported problems in selecting the small icon representing the game associated to a specific artwork. Regarding the content of the games, users judged it quite well (M = 3.73; SD = 1.19) and highlighted the importance of providing games that are easy to solve.

Games were judged amusing (M = 3.67; SD = 1.15) and good for stimulating and improving learning (M = 3.55; SD = 1.37). Among the preferred games, there was the quiz.
Visitors' attitudes to follow the suggestions provided by the system were assessed through a log analysis. The results reveal that only 31.5% of the suggestions were actually followed, and this can be interpreted as evidence that users tend to visit museums quite autonomously, typically without following a straightforward strategy.
All in all, the application was considered useful, interesting and with good potential. Users recommended to improve the localisation precision and performance, and to simplify as much as possible the interaction. Some users suggested to increase icons size and pictures resolution, to ease items selection and to better support the association between virtual and real artworks. Two users would have appreciated the dynamic enabling of the available games, in order to let them play only the games related to visited areas (i.e. the games they would be able to solve by benefiting from the museum visit).

2.3 Accessibility

This subsection reports on a solution for providing support to the blind using mobile museum guides. The main strategy is to exploiting the haptic channel as a complement to the audio/vocal one. The overall goal is to improve the autonomy and social integration of blind visitors. An iterative approach has been followed, in which the proposed system has gone through various usability evaluations and further refinements. The final solution includes vibrotactile feedback enhancement for orientation and obstacle avoidance. Unobtrusive actuators are applied to two of the user's fingers. An electronic compass and an obstacle detector are connected wirelessly to the mobile guide. The study indicates that vibrotactile feedback is particularly useful to provide frequent unobtrusive indications on dynamic information, such as the level of proximity of an obstacle or the angular distance from the right orientation.
Technologies, in addition to new services and products, also provide new opportunities to allow users with special needs, such as people with disabilities, to perform activities previously impossible or particularly difficult to do (e.g. accessing digital information for blind people). The increasing potentialities of mobile devices (smartphones and PDAs) make them suitable for use in assistive technologies or in developing supporting tools (see for example [70]). However, to achieve concrete results, accessibility principles should be applied when designing and developing such products and/or services.

Accessibility is a general term used to indicate that a product (e.g., device, service, environment) is accessible to as many people as possible, including those with disabilities. This is a fundamental feature for systems to allow users with different abilities to use them. In this perspective, a multimodal approach can represent a valuable way to support various interaction modes, such as speech, gesture and handwriting for input and spoken prompts. Thus, by combining various interaction modalities, it is possible to obtain an interactive interface suitable for users with varying abilities.

A well-designed multimodal application can be used by people with a wide variety of impairments. Visually-impaired users rely on the voice modality with some keypad input. Hearing-impaired users rely on the visual modality with some speech input, and so forth.
In this regard, museum environments have been chosen as a domain for investigating how to design and implement a multimodal mobile application that can be easily used by blind people in this domain [74]. The aim of our study is to provide blind visitors with greater autonomy. Even if the blind cannot actually see museum items, visiting an exhibition autonomously is a good way to integrate the vision-impaired into a group of people (e.g. family or friends) and is more effective than obtaining cultural information from a Web site.
An investigation about how the haptic channel, in conjunction with the audio/vocal one, can provide better support for the use of mobile museum guides for blind users, is presented in this subsection.
Blind or visually impaired users must rely upon senses complementary to sight in order to perceive information, and haptic interface technology allows building tangible data surfaces to provide an additional modality for exploration. Unfortunately, the amount of information perceivable through touch is much less than that perceivable through vision. Consequently, multimodal approaches should be deeply investigated in order to enhance the perception of people with visual disabilities.
Some related work is discusses in the following, and the key features of the proposed solution are introduced. The various versions that have been developed and the associated user tests are reported as well, and the results of such tests are discussed as they drove the evolution of the work up to the final version.

The use of haptic output for mobile users has been considered in several studies. An array of 9 tactile actuators making up a wearable vibrotactile display was studied in [81]. Brewster and others [24] proposed Tactons, i.e. structured vibrotactile messages carrying complex information: they studied how haptic feedback alone could encode 3 different parameters (Rhythm, Roughness and Spatial Location) exploiting several vibrotactile actuators. The above-mentioned works highlight the potential suitability of their proposals with mobile devices such as PDAs, but the reported user tests are actually limited to stationary environments.
An evaluation of tactile output supporting mobile interaction is provided in [20], which highlights the benefits achievable through haptic feedback. The tests revealed that user performance significantly improves when haptic stimuli are provided to alert users about unwanted operations such as double clicks or slips during text insertion.
The previously cited works focus on the advantages of exploiting the haptic channel as a complement to the visual one and do not tackle solutions for blind users.
Some proposals for supporting blind users' mobility can also be found in literature. In [3] a haptic direction indicator prototype is proposed to support the visually-impaired in emergency situations. Studies of user requirements indicate that in specific situations, such as emergencies, the supporting device should be small so as to be easily held in the hand. A general purpose navigation system has been proposed in [96], which adopts tactile perception to inform the blind user about the

distance to an obstacle. The authors claim that using multiple sources of vibration to convey information about the environment is more effective than audible feedback. Variable and synchronized vibration pulses have been used to enhance sense of orientation and distance for the user. The navigation system is based on sonar sensors, an embedded micro-controller system and an array of vibrotactile actuators. To convey information to the user exploiting the sensitivity of the hand, the authors combined all three tactile perception parameters: the location of the active vibrotactile actuator, the intensity of the feedback, and pulse duration. However, the proposed hardware seems to be a stand-alone device without any possibility of adapting it to other applications (e.g.: customizing the output of a mobile guide).

The recent progress of handheld computers and mobile phones has enabled the development of compact wearable aid systems for the blind, even in combination with RFID technologies. Possible applications are related to indoor solutions to support visually-impaired people in mobility and orientation. [35] is a guidance system, based on a cane with embedded RFID reader and a Bluetooth module. Sensed data are sent via Bluetooth to the handheld device, connected to a remote server, that guides the user by means of speech-synthesized instructions. This solution is based on a general purpose handheld device, which requires blind users to follow predefined (tagged) paths, thus limiting the user's freedom of movement. [30] aims to help them in finding the shortest path to a destination, as well as in the case they get lost. The proposed system embeds RFID tags into a footpath that can be detected by an RFID reader with a cane antenna. The dedicated device is portable and equipped with a headphone for navigation where only voice (in mp3 format) is used to guide the users. The system however does not include any obstacle detector.

The system discussed here, combines RFID-based localization with a compass, with the aim to provide information on the current user position and to determine heading; a distance sensor is also used to provide information on the distance from stationary and/or moving obstacles. Another RFID-enabled navigation for the blind has been proposed in [108]. Detected tags provide the coordinates of their location, as well as other information, and orientation is supported by vibrotactile output. While the system surprisingly does not depend on a centralized database, it supports navigation through predefined paths marked by RFID tags, like RadioVirgilio/Sesamonet. [26] is an original solution for providing blind users with audio information about RFID-tagged objects nearby (such as temperature and weight). The RFID reader is embedded in a glove to let the user freely explore the area.

[32] also describes an assistive system exploiting electronic markers to provide useful information to the visually impaired. Tagged objects are detectable by a mobile device that provides descriptive information. Tomitsch and others [103] have exploited audio-tactile location markers (ALMs) combining audible signals and tactile identification for making real-world tags accessible for users. Real objects are marked by passive Near Field Communication (NFC) tags. Since such kind of tags are activated at low ranges (below 10cm), Bluetooth technology is used to locate them from greater distances. An audible signal allows to identify the position of the tag, when a mobile device (i.e. cell phone) is detected in the

neighbourhood through Bluetooth exploration. Although this solution includes both auditory and tactile feedback, it seems to be somewhat expensive. [5] is a proposal for helping blinds in public transportation. Users rely on a PDA or mobile phone with WLAN or Bluetooth connectivity to activate a stop request or to be informed about the next stop.
While the above reported solutions for the visually-impaired provide information about the surrounding environment, they do not offer support for users freely moving towards the tagged objects and, at the same time, to avoid potential obstacles. In particular, the work discussed in this subsection is specifically aimed at improving the mobility of blind users in this regard.

2.3.1. Key features of the solution

A combination of various technologies enables the multimodal user interface for supporting blind users. Broadly speaking, the type of context of use addressed by the proposed system is a visit in an indoor environment with various points of interest, such as a museum. In this perspective, the components f the prototype have been designed to support (1) user movements and orientation, (2) obstacle avoidance along any given path, and (3) access to information and descriptions of items on display (such as museum artworks). The proposed system is composed of a PDA equipped with technologies that address the goals mentioned above.
Concerning the first goal (i.e. user movements and orientation), the system adopts Active RFID technology. In order to facilitate object detection in a rather large environments, long-range tags have been chosen, as they are detectable within 5 meters. A compact flash (CF) reader has been plugged into the PDA in order that the mobile device can be easily held in one hand. To enable suggestion of the appropriate direction for moving towards the targeted object (i.e. next artwork), an electronic compass has been specifically designed for this application. The wireless connection between the PDA and the proposed wearable compass is Bluetooth-enabled. The system has also been equipped with vibrotactile feedback for supporting the user in orientation. Different intensities and durations of vibration encode the indications to be provided to the user to achieve the right direction.

A distance measuring sensor has been embedded into the electronic compass box, to enable obstacle detection. Both the compass and the distance sensor are connected, through a microcontroller, to the same Bluetooth module, which the user device connects to.
The distance sensor allows the mobile application to inform the user when s/he is approaching a potential obstacle. The kind of feedback has been modified over the various versions that have been developed. Both sound and vibrotactile sequences were investigated.
Regarding the vocal description of the artworks and the messages provided by the system, a solution based on real-time speech generation has been chosen instead of MP3 recordings. To this purpose, the TTS (Text-To-Speech) engine used, which comes with multi-language support, offers several benefits, such as requiring less storage space (while MP3 files would require much more memory than texts) and minimal effort to modify messages and descriptions. It is worth noting that, in the event of content changes, no recordings have to be made again, and this is an important aspect especially when dealing with multi-language applications.

As mentioned above, although generating audio messages on the fly is an easy task, vibrotactile output could be a reasonable way to convey feedback in some cases. One of the reasons is the unobtrusiveness of such technology, which might be more appropriate than vocal output in public places.

2.3.2. First prototype (RFID, electronic compass and audio/vocal feedback)

2.3.2.1 RFID-based localization

The first prototype was characterized by a localization infrastructure based on a number of RFID tags deployed throughout the exhibition area: each tag was placed by an artwork [47]. The correspondence between artworks and tags is stored in the museum database, as well as the position of each artwork within the room. It is worth pointing out that, in the accessible museum guide, artworks position is essential for suggesting the right direction to the user.

The schema in Figure 12 summarizes the architecture of this first prototype.

Bottom-up in the schema:

- *RFID-Reader* is the hardware that detects the RFID tags in the environment, i.e. a CF (Compact Flash) card plugged into the slot of the handheld device, provided by Identec Solutions (http://www.identecsolutions.com). Each tag transmits its ID with a constant power level that enables detection within 5 meters. For each tag the RFID reader also detects its signal power (that is, the RSSI – Received Signal Strength Indication, which depends on how far the tag is), reporting it to the software layer. A location event is triggered whenever a new RFID tag is detected or the signal strength of a tag has changed.
- *TagDetector* is the software embedding the RFID monitoring thread that generates location events. It interfaces with the RFID reader through pre-compiled software modules provided by the hardware manufacturer.
- *Electronic Compass* is the device for sensing user direction in absolute orientation degrees. The compass is needed because calculating the user motion vector based on the signals detected from the RFID tag network is rather problematic. Since none of the existing commercial solutions seemed to be suitable for a mobile application, the device has been expressly designed and developed to meet the requirements. It consists of an analog compass sensor and a microcontroller that manages Analog to Digital Conversion (ADC) and data serialization. The compass device is battery operated, has an embedded Bluetooth interface with SPP (Serial Port Profile) and is detected by the PDA as an external wireless peripheral. The small size and low weight of the compass device make it easily wearable. Figure 13 illustrates a blind user carrying the PDA and wearing the compass as a necklace.
- *UserDir* is the interface software to the compass device. It reads and filters the stream of values coming from the compass and computes the direction 3 times per second as a value in the interval [0,359] degrees with respect to the North.
- *Museum DB* is an XML description of the whole museum: authors, artworks, sections and associations between artworks and RFID tags. These resources make up a simple GIS (Geographical Information System) that holds enough information (sections geometry and artworks positions) for supporting a blind user. The access to the DB is triggered by the Map Event Handler.

- *Map Event Handler* is the module that catches the events triggered by the TagDetector and asks the UserDir for the current user direction. This module also queries the Museum DB for the environment topology (to compute the paths) and for artwork information (to provide automatic description upon reaching the target artwork).
- *Visit* supports automatic access to museum info such as artworks and section descriptions.
- *VUI* (Vocal User Interface) exploits an embedded TTS engine, provided by Loquendo (http://www.loquendo.com), which synthesizes the speech for describing artworks/sections and for giving direction tips on the fly ("… rotate left…", "… carry on in this direction", "Please, stop!").

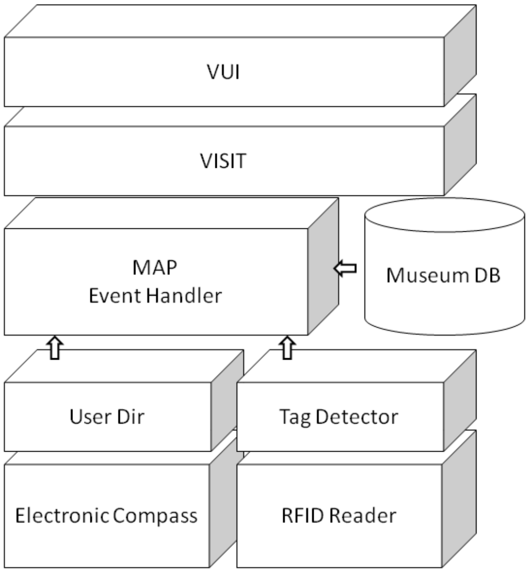

Figure 12: Architecture schema of the accessible mobile guide.

As soon as an artwork has been reached by the visitor, the event handler launches a vocal message that informs him/her about which item is being visited. If the user accepts, then the description is played, otherwise the path to the next artwork is computed. User input is conveyed through the keypad of the PDA, since the physical buttons are easily distinguishable due to their shape.
The compass information is used in conjunction with RFID detection. When the user requests help to reach the next artwork , the guide computes the line joining the currently visited artwork and the target one. This is possible because the application is aware of the environment topology, which is stored in the DB. Initially, the current user position (detected by RFID) is approximated to the starting point and the direction is sensed in real time by the compass. Thus, the direction is updated many times per second in order to provide the user with continuous indications: by this way s/he can more easily keep the right heading towards the target not to stray from the best path.

Figure 13: A blind user carrying the early prototype of accessible mobile guide (PDA in the right hand and electronic compass round the neck).

In the early integration of the compass support vocal instructions were used to guide the user. Depending on the direction change needed to reach a specific location, the application vocal support synthesizes speech such as “turn X degrees left/right” (with X ≤ 180). When the right direction is reached the user is asked to move towards the artwork by the instruction: “slowly approach the artwork continuing in this direction”. As soon as the user enters the destination area (about 5 meters around the artwork tag) s/he receives a vocal notification and a “beep” starts to sound repeatedly. The frequency of the beeping depends on the current distance from the destination tag: the faster the beeping, the closer the target artwork. Distance monitoring has been considered as the only way to ensure the user reaches a point of interest. It is worth noting that the compass support is useful to direct the user towards the artwork, but the system has no information about the path actually taken.

2.3.2.2 Evaluation of the first prototype

Before carrying out a full user test, a preliminary experiment on the very first prototype was conducted in order to set up the version to be used. In the preliminary evaluation, two blind users were involved to collect useful comments and data while using the system. They were initially instructed to reach an artwork by exploiting only the distance support, that is the repetitive sound indicating how far the destination tag was. Both users accomplished the task, but they took quite a long time. Users reported that looking for an artwork placed many meters away from the current one is a too complex task if no direction aid is provided. This is due to the need for scanning, in the worst case, almost the entire room before reaching the suggested destination. A second session trial was performed on the guide version enhanced with the compass module. Users were able to discover the proposed artwork location and arrived very near the associated tag. The compass support was judged essential and some improvements were suggested. The first user would have preferred a continuous sound with variable frequency rather than vocal tips: he stated that degrees-based tips are not intuitive. The second user, on the contrary, appreciated the vocal aid but pointed out that not everyone is familiar with measurements in degrees. She suggested indicating the direction by simple sentences such as “rotate left/right” or “turn back”. Therefore, a test of these two kinds of audio feedback with a larger group of people was planned in order to evaluate their real impact on the blind. All in all, users found the orientation support to be a reasonable way to help the blind in visiting a museum independently.

Keeping in mind the observations gathered from the two users, a user test involving a larger number of blind users was performed. Two versions of the prototype with the two different audio feedback were used in this evaluation. The main goal of the test was to compare the usability of the two prototype versions with different types of audio feedback. More specifically, the evaluation was targeted at answering the following questions: (1) Can an electronic compass support a museum visit? (2) What is the most appropriate audio feedback for a blind user?

For these purposes, the evaluation regarded two versions of a guide prototype differing from that preliminarily tested only in the type of audio feedback. The first version adopted simplified vocal tips such as “rotate left a bit” and repetitive “beep” sounds with variable delay to indicate distance. The second version used a continuous sound with variable frequency to suggest direction and repetitive “beeping” sounds with variable frequency to signal the distance. In the latter version, the frequency of the direction sound decreased as the right orientation was achieved. Once aligned, on the contrary, the distance sound frequency increased as the user approached the target.

Five participants were recruited: four of them totally blind from the birth and one with a little residual vision. The age ranged from 33 to 69 years. Four of them used a computer with a screen reader in Windows environment daily. None of them had any experience with PDAs.

Users were asked to reach a specific artwork by exploiting the compass-based guide. Since the experiment was also aimed at analysing the kind of feedback, each user tried to reach different artworks by using the two versions of the guide prototype. In order to avoid a possible bias due to the learning process, three users used the vocal version first and then afterwards the one based only on the sounds; whereas for the others the process was inverted. The task assigned was to start from a specific artwork to reach another one. The artworks to reach were different for the two guide versions. The users were observed as they performed their tasks. The start and finish time was recorded for each user and for each task. At the end of the tests, users were asked to fill in a questionnaire aimed to collect subjective comments and recommendations.

All the users were able to accomplish the assigned tasks. The recorded time for each user for all tasks revealed a significant time saving in reaching the artwork by using the version with both vocal and sound support. Just one user spent more time using the full version. For the other users, the time saved ranged from 50% to

82% (M = 35.57%). Users stated they encountered some difficulties in carrying out the tasks due to the system response time, i.e. the time lag of the direction tips. This aspect has been subsequently improved by better filtering the compass event stream.
Concerning subjective opinions, the users could express a value from 1 (the most negative value) to 5 (the most positive value) with respect to various aspects. They declared that the RFID technology can be a useful support (M = 3.6; SD = 1.6) not only for localization, but also for daily activities. However, some of them reported that this methodology should be integrated with other technologies in order to be more precise and reliable. On the other hand, regarding the use of an electronic compass in a museum context, the users declared that such a support would be a useful assistance to allow a blind person to freely move among the artworks (M = 4; SD = 1.2).
Most users (4 out of 5) preferred the version with vocal and sound feedback, while one user reported that using only sound feedback would have been better because sounds might be more intuitive and less annoying. This can be easily explained by the fact that he is an expert in electronic devices. The other users preferred the version with vocal and sound support, declaring that such support is more intuitive especially for a non-expert who is using the system for the first time. In fact, the proposed prototype is designed to be used in a museum context where visitors have no time to learn and become familiar with the guide. Regarding the sounds, one user suggested using different kinds of alerts (e.g. just three types) and well recognizable (i.e. they should have very different frequencies) because very similar sounds could not be easily differentiated. In general users appreciated the support for their autonomy of the application. They provided some hints to improve its efficiency, and one user suggested the addition of tactile feedback, similar to that of mobile phones.

2.3.3. Second prototype (additional vibrotactile orientation feedback)

2.3.3.1 Haptic support

The second prototype aim was to evaluate possible benefits coming from the addition of haptic feedback [48]. The main goal was to investigate whether using multiple sources of vibration about the environment is more effective than audible feedback alone to convey information. The haptic output module, which is an add-on specifically designed and tuned, consists of a plastic box slightly thinner than a packet of cigarettes. The box is attached to the back of the PDA and contains circuitry able to detect infrared signals from the PDA. The photodiode (i.e. infrared detector) protrudes from the box so that it is aligned to the infrared port of the PDA. The battery-operated circuitry also drives two vibrotactile actuators according to the commands sent by the PDA via the infrared interface. Each motor is attached to a rigid surface of about 1x1 cm and is connected to the box by a 10 cm wire. The motors can be fixed to the index finger and thumb by Velcro strips to let the rigid surfaces transmit vibrations to the fingertips. The motors have been separated from the box to facilitate distinguishing the channels. Otherwise, if motors were attached to the box it would have been very difficult to insulate them and the vibration of a single motor would have propagated to the rest of the device, making harder for the user to distinguish the vibrating side.

Figure 14. One of the actuators used in the experiment (top left), the PDA fitted with the haptic module (top-right), the mobile equipment fitting on the user's hand (bottom).

The actuators are RMV (Rotary Mass Vibrator) motors like those that enable vibration in many mobile phones (see Figure 14 top-left). The driver circuitry controls each motor independently. Each command sent by the PDA encodes the state of the haptic interface, that is, for each motor, the on/off flag and the vibration intensity value. The flag controls the switch (i.e. a transistor) while the intensity value, a byte, is passed to a 256-step Digital Programmable Potentiometer (DPP). The DPP is a MCP41010 from Microchip Technology Inc.

The haptic module circuitry is managed by a 4 MHz microcontroller whose routine is able to decode a command and to upload the motor states up to 60 times per second. The microcontroller is a PIC16F84A from Microchip Technology Inc. and has been programmed in assembly language through the MPLAB Integrated Development Environment provided by the manufacturer.

The communication between microcontroller and each DPP is performed via Serial Peripheral Interface. The DPP comes with embedded SPI, while the microcontroller does not have built-in SPI capability. Thus SPI was thus emulated by a routine developed for the assembly program of the microcontroller. The interior of the vibrotactile main module, together with the circuitry, is shown in Figure 15.

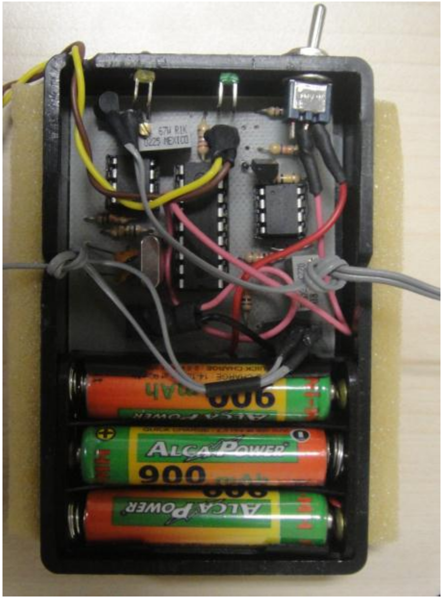

Figure 15. The interior of the vibrotactile module.

To change the haptic device state the PDA software application has to generate and send a 3-bytes command with the switch flags, the intensity values and some check bits. For example, a haptic message like a short, intense left vibration requires two commands. The first, with left flag = 1, right flag = 0, left speed = 255 (the maximum value) and right speed ignored initiates the left engine fast vibration. The second command, with left flag = 0, right flag = 0, left/right speed ignored stops the motor. The vibration length depends on the delay between the start and the stop commands. A complex haptic feedback such as a left-right fading can be created by repeatedly sending commands where the left speed parameter decreases and the right one increases (or vice versa). Since the latency between the *sendCommand* function call and the new motors configuration is about 15 ms, it is reasonable to assume that even more complex effects may be created, such as the rhythms discussed in [24].

2.3.3.1.1. Orientation and Haptic Feedback

One of the aims of this stage of the study is to reduce the time and effort required by the blind user to move towards the next artwork. Indeed, while localization is ensured by a grid of RFID tags and is provided by messages such as “you are approaching the artwork x”, orientation (enabled by the electronic compass) is key support for successfully reaching the destination area.

In this application, usability is related to the time needed to reach the next artwork. Even in the smallest museums or exhibitions there are tens of artworks/items. The user should consider the guide as effective, i.e. as an alternative to the human companion. If not, the guide would be perceived as useless, turning the museum visit into a frustrating experience.

As already mentioned, haptic output has been investigated according to the suggestions made by some users of an earlier test. It has been considered as a complementary (rather than alternative) modality for supporting blind users' orientation.
It is assumed that the museum visits are usually made just once by visitors, who probably do not want to spend much time familiarizing themselves with the interface. For this reason, and taking into account the suggestions by users in a previous test, only a small amount of information has been encoded in the haptic feedback. Four types of patterns have been configured corresponding four vocal suggestions of the application: "Rotate Left", "Rotate Right", "Rotate Left a bit", "Rotate Right a bit". Rotation direction (left/right) is given by Spatial Location of the activated motor (i.e. which finger is vibrating). Rotation angle is indicated by Duration and Intensity of the vibration: a strong and long (2 seconds) impulse or a light and short (700 ms) one to indicate whether rotation must be more or less than 90°, respectively.
Haptic patterns provide an indication of the distance to the target as well: once the user has aligned to the best direction and has reached the destination area, short vibration pulses are activated on both sides. The delay between pulses reduces as the destination tag signal grows (i.e. the user approaches the tag). Feedback on the distance may help the user in reaching the target even if the followed direction is slightly different from the ideal one.

2.3.3.1.2. Vibrotactile patterns

Figure 16 summarises the architecture of the haptic support and shows how it fits within the general one. The mobile application exploits a one-way infrared connection (IrConn) to communicate with the haptic device. Sending a command is a trivial operation since it simply consists of writing bytes on the infrared port stream. As already mentioned, in order to play a basic haptic message two commands must be sent to start and end the vibration. The application has to manage the vibration duration as well. As output complexity grows - i.e. many custom vibrating patterns are needed - a mechanism to arrange and execute them is highly desirable.

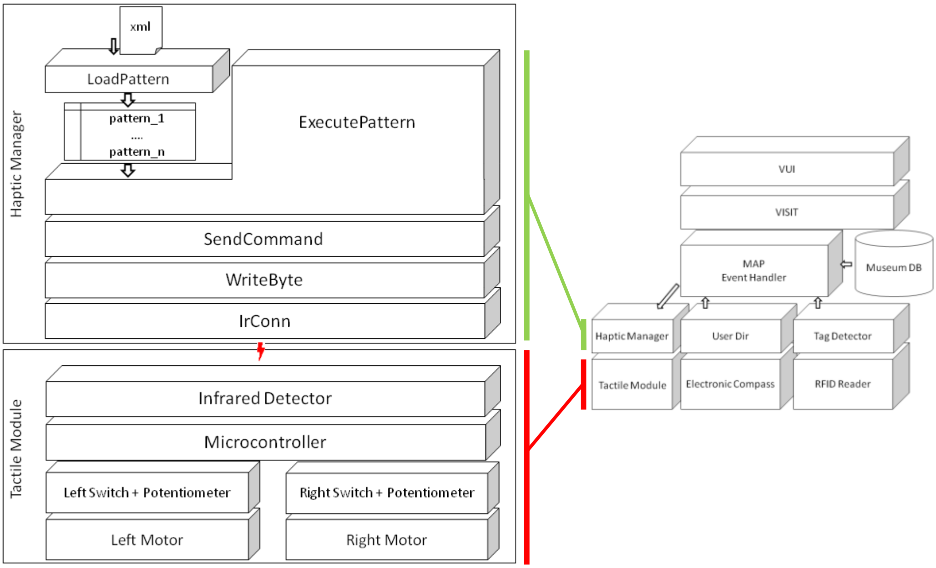

Figure 16. Hardware and software layers for the haptic framework (left side) and how they fit on the overall architecture (right side).

The development of such architectural framework was performed simultaneously with the creation of a tool for editing custom vibrotactile patterns through the PDA touch screen (see Figure 17). The pattern of the vibrotactile output for each side (left/right) is defined by dragging the curve, which is associated with the vibration intensity over time. By choosing size (number of steps) and delay it is possible to define the pattern granularity:

- *Selected side* radio button indicates which actuator (left or right) is currently active for editing. The curve points of the active actuator are represented by blue squares, while for the inactive one the squares are light-grey.
- *Steps* is the number of points on the curves.
- *Delay* is the pause between each pair of consecutive points on the curve and determines the duration of the vibration impulse. The number of possible intensity values is a fixed value since it depends on the hardware capabilities of the potentiometers.

Figure 17 shows three patterns that differ in shape as well as in the number of steps (granularity) of the curve. Pattern (a) represents a fading on the left side and a continuous vibration on the right. The pattern is encoded in 28 steps with a delay of 35 milliseconds. Pattern (b) is a high intensity vibration with some roughness only on the left side (right actuator is off). Roughness is generated by lowering the intensity for extremely short periods (the delay is 25 ms). Pattern (c) is a left to right fading in 52 steps with a 15 ms delay. Increasing the steps and decreasing the delay enables refinement of the vibrating pattern.

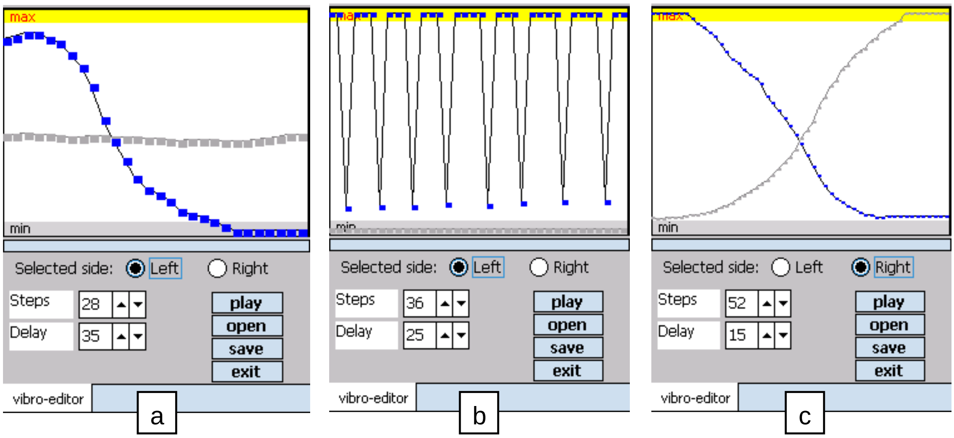

Figure 17. Three examples of vibrotactile patterns editing on the graphic environment.

The vibration intensity values for both sides can be stored in an XML file containing the delay value (which is constant and indicates the pause between consecutive intensity values).
The highest layer of the HapticManager module (see Figure 16) is able to load vibrotactile patterns defined in XML files. Loading is performed during application initialization.
Loaded patterns can be executed at run time by passing the appropriate index to ExecutePattern. Execution consists of a cycle that reads data structures content and sends commands to the device. Such a cycle should be performed in real time to preserve the peculiarity of the curve. One possible strategy is to give high priority to the thread that executes the cycle, but the fundamental requirement is to parse the XML file only once, that is at loading time.

2.3.3.2 Evaluation of the second prototype

The additional user test conducted with a group of blind people was aimed to evaluate the second prototype version and particularly the possible benefits coming from the haptic feedback. The evaluation concerned two subversions of the guide prototype that differ from that previously tested. The previous evaluation revealed that the vocal version using clear messages was preferred by the users with respect to audio sounds. Furthermore, subjective opinions indicated that a vibrotactile output could have been an appropriate way to convey feedback.
Thus, for evaluating this second prototype, the subversion based on vocal messages alone was compared with another new one using vibrotactile feedback for orientation. In the solely vocal subversion, tips such as “rotate left a bit” and repetitive “beep” sounds with variable delays were used to indicate distance and guide the user towards an artwork. The vibrotactile subversion used right and left vibrotactile feedback to suggest direction and a double vibration corresponding to the repetitive “beeping” sounds with variable frequency to signal the distance. The vocal support provided in the vibrotactile subversion regarded just the audio descriptions of the artworks. The users had to hold the PDA in the left hand and,

for the vibrotactile version, to wear the two motors on the thumb and index finger (see Figure 14). The choice of the index finger rather than the middle finger is based on the sensitivity level [66].

The goal of the test was to analyse whether there are any differences in terms of user performance between the two different feedbacks. To this end, each user tried both subversions to carry out the assigned tasks. An observational method was used to follow the users testing the prototypes. Both quantitative and qualitative data was collected through the conducted test. First, in order to analyse the differences between the two subversions, the time taken by each user to carry out the tasks was measured. Secondly, after the test each user was asked to fill in a questionnaire to gather subjective information and possible suggestions. The logged data were used to compare the effectiveness of the two subversions. Indeed the time spent to perform each task was used as a quantitative measure to compare the two types of feedback.

Eleven blind participants were recruited for the test: 7 women and 4 men. The age ranged from 27 to 66 years. All of them had used a screen reader in a Windows environment before. Six of them had previously used a PDA.

Users were requested to reach a specific artwork by exploiting the multi-modal guide capabilities. Each user tried to reach different artworks using the two different prototype subversions since the experiment was mainly aimed at analysing the kind of feedback. In order to avoid a possible bias due to the learning process, five users used the subversion with vibrotactile feedback first and then afterwards the one based only on vocal feedback; whereas for the others the order was inverted. The task assigned was to start from a specific artwork to reach another one. The artworks to reach were different for the two guide subversions. All users performed the same tasks. Each test was carried out separately in order to avoid user bias caused by observing other tests. Users were observed while they performed their tasks. As mentioned, the starting and finishing time was recorded in a log file for each user and for each task.

Every user was able to accomplish the assigned tasks and the time log for each task allowed to compare the time spent in carrying out the task when using the guide with the two different types of feedback. Compared time data revealed time saving in localizing the artwork by using the subversion with vibrotactile orientation support, though the difference was not statistically significant.

Data recorded for each user was examined in order to measure the effective time saving. The analysis was performed taking the users as the statistical unit and considering the time(s) taken by each user to accomplish the task.

Parametric analysis was applied according to two tests: Exact Kolmogorov-Smirnov ($N = 11$; $Z = 0.835$-0.906; $p = ns$) and Levene's test ($F=1.179$, $p=ns$). A Paired T test ($N = 11$, $t = 0.463$, $df = 10$, $p = ns$) revealed a non-significant difference in the time required to complete all tasks by using the two subversions (vocal and vibrotactile). This means that the two prototype subversions were similar and more or less equivalent for the purpose. Users declared that they felt more confident with clearer vocal messages, especially about the direction. Probably, in order to become more familiar with the vibrotactile feedback, it is necessary for blind users to experience it for a long period. Unfortunately, this is not the case, because our system is conceived for museum visitors who prefer short time for training.

Concerning subjective opinions, the users rated various aspects on a scale from 1 (the most negative value) to 5 (the most positive value). One aspect regarded the preference level for the two kinds of feedback. The expressed preferences revealed there is no significant difference between the two types of feedback. According to the Exact Kolmogorov-Smirnov ($N = 11$; $Z = 0.780$-0.850; p = ns) and Levene's test ($F = 0.508$, p = ns), a Paired T test revealed a non-significant difference ($N = 11$, $t = 1.047$, $df = 10$, p = ns). This result confirms the quantitative information gathered through the time logging. On average the users rated the vocal feedback well ($M = 3.81$; $SD = 0.98$) and the vibrotactile version slightly less well ($M = 3.18$; $SD = 1.47$). Various suggestions on how to improve both versions were expressed through the questionnaires. One user stated that vibration intensity range should be made wider by increasing the motors' peak speed. Another would have preferred different vibration rhythms rather than variable intensities/durations to signal the angle of rotation. Two suggested increasing the updating frequency of tactile messages, in order to speed up the orientation process. Three users declared that an obstacle detection functionality would allow the avoidance of physical barriers, improving the autonomy of blind visitors.

2.3.4. Third prototype (obstacle detection support)

Following the criticisms and suggestions provided by the users that tested the second prototype, the solution was modified. The detection hardware was enhanced by a distance sensor to alert the user about obstacles on the path. The module was also modified to be carried on a belt rather than through a necklace, because previous experiments highlighted that the direction sensing was affected by user's movements when the compass device was carried round the neck.

2.3.4.1 Obstacle detection

Obstacle detection is enabled by a Sharp-GP2D12 sensor embedded on the compass device box (see Figure 18). Obstacles are detected when they are between 10 and 90 cm from the user and within a cone of about 35°. Since the distance (as well as the orientation) is sampled at about 30 Hz, the upload latency is less than 40 ms. Thus, the mobile application is able to warn of approaching obstacles such as walls or other people in quite real time.

While direction aids are given by left or right side haptic messages with intensity and duration depending on the turning angle, obstacle distance is provided by the same pattern on both sides. Two strength levels of feedback have been considered for each direction aid and 3 levels for obstacle alerts. The feedback strength level is given by the intensity of the impulses making up the pattern and the duration of the pattern. Intensity and duration increase as the obstacle distance decreases or as the distance from the right direction increases. The vibrating sequence is made up by reproducing the same pattern with a delay between two consecutive patterns. For obstacle feedback the delay is about 200 ms. Pause time between direction aids is variable and is typically much longer. It depends on when the direction of the user is stable (i.e.: when the user stops turning). This is done by the direction aid process that continuously checks for the current user direction and provides the next aid (if needed) when the direction is sufficiently stable (i.e. its variation is under a certain threshold for a while, about 3 seconds).

In particular, 7 patterns have been defined that are briefly explained in Table 1.

Pattern vibration strength		Vocal comment	Meaning
Left side	Right side		
low	OFF	"Rotate left a bit"	Rotate left less than 90°
high	OFF	"Rotate left"	Rotate left between 90° and 180°
OFF	low	"Rotate right a bit"	Rotate right less than 90°
OFF	high	"Rotate right"	Rotate right between 90° and 180°
low	low	none	Obstacle distance less than 30 cm
medium	medium	none	Obstacle distance between 30 and 60 cm
high	high	none	Obstacle distance between 60 and 90 cm

Table 1. Correspondences between vibrotactile patterns and vocal comments.

Each actuator is characterized by a certain strength in each pattern (see Table 1), which depends on the duration and the structure of the pattern. Note that the vibrotactile direction aids (vocally conveyed by "Rotate left/right …" sentences) only provide vibration on the side to which the user has to turn.

Two subversions of the support have been tested: one providing continuous obstacle feedback and another one emitting feedback only when the obstacle distance reduces (i.e. when the user moves towards it). Users were also asked for their opinion about the vibration sequences.

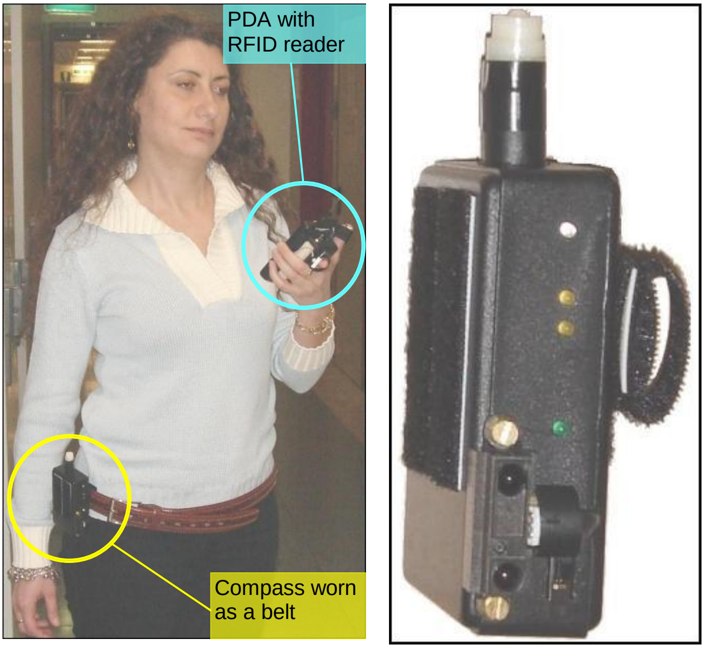

Figure 18. Left: a blind user carrying the mobile guide in the left hand and the orientation/distance detector as a belt. Right: detail of the compass device. The white cylinder on the top is the compass sensor; the distance sensor is located on the bottom-left.

2.3.4.2 Evaluation of the third prototype

The early investigation on vibrotactile feedback for orientation purposes was devoted to determining whether users preferred vocal sentences or haptic messages as orientation aids. Although no significant preference emerged, the vocal and haptic channels have been combined to indicate the direction to take. Their combination, indeed, seemed to provide more information. According to the observations made by some users, obstacle detection has been integrated into the system to signal obstacles in front of the user by means of haptic feedback. Custom sequences, with increasing vibration intensity when approaching any obstacle were produced on both sides (i.e. fingers).

The latest user test was aimed to evaluate two versions of the vibrotactile feedback for obstacle detection: one with continuous vibration, whose intensity depends on the obstacle distance, and the other providing vibration only when the distance to the obstacle decreases (i.e. when the user is approaching an obstacle).

A group of 7 blind users, 3 women and 4 men, aged between 25 and 40, was involved. The simulated exhibition environment was set-up in an indoor area.

Each user tested both versions of the obstacle avoidance feedback. The two trials consisted of reaching a target (simulated) artworks while coping with some obstacles in the environment. The two target artworks were different. All users wore the obstacle detection in a belt on their left hip, held the PDA in the left hand, and had the vibrating motors attached with small Velcro straps on the same fingers

(thumb and index), and started from the same location. In order to avoid any bias in the learning process, half of the users tried the continuous feedback version first and then the discontinuous, while the order was inverted for the others.
Time taken to perform the requested tasks was logged for each user. The time spent served as an index to compare the effectiveness of the two versions. All the users were able to perform the tasks. At the end of the test, they were asked to compile a questionnaire with subjective considerations and possible suggestions. Each version of the haptic support was rated on a scale from 1 (the most negative value) to 5 (the most positive value).
According to the rating (mean = 4.00 - 3.28, SD = 1.25 - 1.15), the version with discontinuous vibration was rated higher, though the difference was not statistically significant. Due to the small size of the samples ($n < 10$), a nonparametric tests [99] was applied. The normality could not be verified due to the small sample size. The exact Wilcoxon sign test was devoted to assert differences in the time taken by the task (by the same users) and in the preference score expressed by each user for the two versions ($k = 2$; $n = 7$).
Users that preferred the discontinuous version described it as less annoying than the continuous feedback, especially when standing in front of an artwork or a wall, which are implicitly obstacles. Probably due to the small size of the sample, the Exact Wilcoxon test has returned $p=0.12$, thus it might not be asserted any trend.
On the other hand, 4 of the 7 users preferred the continuous feedback, indicating a slight preference for such a support overall. In the questionnaire, this choice was informally motivated by the “safety” users felt with a continuous indication about an obstacle. The preference is also confirmed by the task duration: the mean times taken to reach the targets were 139 (SD = 50.95) and 155 (SD = 54.71) seconds for the continuous and discontinuous vibrotactile feedback, respectively. The difference of time spent for performing the task between the two versions is significant ($p < 0.05$). Users spent less time to achieve the task by using the continuous vibrotactile prototype.
Additional observations were also collected: most users reported that the system latency in providing aids needs to be reduced. Other suggestions regarded possible improvements in the hardware positioning so as not to obstruct user’s movements: line of sight of the distance sensor carried on the hip was sometimes obstructed by the user arms.

2.3.5. Final prototype

Based on the comments gathered from the previous user study, some changes were made both to the wearable hardware and to the user interface. The box of the compass/distance module has been slightly modified to easily attach it to the front side of the belt (see Figure 19). Therefore, distance detection is not affected by user’s arm movements and heading detection is not affected by unwanted shocks. In addition, the distance sensor is actually pointed forward and provides a more balanced detection of the obstacles lying in front of the user. The program embedded on the microcontroller of the detection module has also been improved. While the previous version just continuously transmitted the sensed values (i.e. heading and obstacle distance) to the PDA, the current version transmits only as needed. That is to say, whenever the application needs to refresh the parameters’ values, it queries the detection module for the current ones. This polling strategy has been set up for energy saving purposes: limiting the data gathering from the sensors and the transmission over Bluetooth increases battery life (especially for the detection module).

Figure 19. The detector module with improved layout.

In the final prototype the vibrotactile module has been upgraded too: the vibrators have been replaced by two standard electric motors equipped with a small custom rotary mass. Although such motors are slightly bigger than the previous ones, they have a very low start-up current that enables them to start and continue running slowly, providing a larger speed range. The extended speed range improves the customizability of the output which facilitates distinguishing the complex vibrotactile tips.

As suggested by the results of previous tests, which showed no significant difference between the continuous and discontinuous vibrotactile feedback for obstacle detection, a customizability of the vibrotactile support has been added to the application. Now, it is possible for the user to choose the vibration intensity level for the direction indications and the obstacle presence alerts through the PDA keys. The perceived intensity of the vibration is related to the structure of the vibrating pattern. At configuration time, several predefined patterns are created through the graphical editor and deployed on the mobile device. Whenever the user changes the vibration intensity level a new subset is loaded from the set of predefined patterns and stored into data structures. The working subset consists of a few vibrating patterns with different intensity. As in the previous version, the pattern is chosen from among the subset depending on the type of suggestion/alert to provide: strong patterns are used for direction changes of over 90° and for alerts about obstacles in the near proximity (30 centimetres or less).

Another feature added to the final version is the possibility of deactivating the obstacle alert. The user, after approaching the destination item and while listening to its description and/or touching it, may want to switch off the vibration since it would not provide any useful information (the user already knows that s/he is in front of an obstacle).

Thanks to the new demand-response behaviour of the detection module, exclusion of the distance detection results in lower power consumption by the distance sensor (no sampling is required) and the Bluetooth module (30% less data are sent to the PDA) and thus in energy savings in the detection module. An energy conserving strategy has been adopted with respect to direction detection as well: when heading sensing is not needed (i.e. whenever the user does not need the mobility aid), the detection module is not queried for direction.

2.3.6. Discussion

The study discussed was aimed at investigating how haptic feedback can be exploited to improve user interaction as well as to provide additional information for blind users. It has investigated what type of information can be provided through a vibrotactile feedback compared with that coded through audio and vocal feedback. A mobile guide has been considered as a case study for these purposes. Other studies on comparing both audio and tactile feedback have been conducted: [57] discusses how audio and tactile feedback can be effectively used on mobile devices. The authors consider cross-modal feedback in which two modalities are used to provide the same information. The audio and vibrotactile feedback is investigated in a general context, i.e. people with disabilities were not involved. According to the findings, two modalities can be used in a redundant manner with positive results. The study presented in this dissertation, instead, shows positive results when the two modalities are used in a complementary way. The work has provided useful indications about how vibrotactile feedback can be exploited in a multimodal mobile guide for the blind. The proposed solution is unobtrusive because it does not require wearing cumbersome equipment and involves only two fingers of one hand (which is in any case used to hold the PDA). In this way, the support allows users to freely move about instead of requiring them to follow specific paths.

Obstacle detection, that is a further important issue for the blind, has been considered in the prototype. It has been observed, considering also the user comments, that vibrotactile feedback can be a valuable way to provide this kind of information.

The study indicates that vibrotactile feedback is particularly useful to provide frequent unobtrusive indications of dynamic information, such as the proximity of an obstacle or the distance from the right orientation. Vibrotactile feedback is therefore considered to be a good complement to the vocal modality, which would become tedious when too many messages are played, and can be better used to point out specific events (such as reaching a given target). The vocal channel, in the considered domain (mobile guides) is in any case used to provide relevant content, such as long descriptions. Therefore, an additional modality for providing other types of information is useful to avoid overloading the vocal/audio channel. Regarding the lessons learned, it has been observed that providing intuitive support on target achievement via the haptic channel is quite difficult in comparison to a vocal modality. Since the vocal channel can clearly encode a wide variety of messages, vocal feedback is preferable for various kinds of indications, such as “target reached” or “please, turn round”. Whereas, for other types of dynamic feedback, a vibrotactile channel can be more effective. For example, informing the user of the obstacle distance via a haptic channel is effective and unobtrusive. Increasing and decreasing the level of vibration can be a good way to code this

information. In addition, using the haptic channel leaves the user free to dedicate the audio channel to social interaction when desired.
The limit of tactile feedback alone is that it is not suitable for some specific information and that it may require some time for the user to become familiar with its vibrotactile messages.
During the experiments, more complex indications such as “turn round” were sent through the vibrotactile feedback, but this was not appreciated by the users. Vibrotactile feedback seems to be more effective for repeated short information (e.g. the user approaching an obstacle).

One aspect that has been clearly suggested is that the solution for providing the haptic feedback should be customizable, and users should be allowed to select the type of feedback that they feel more appropriate (e.g., continuous or approach-dependent vibration to indicate obstacle proximity).
Furthermore, different users may have different preferences regarding the feedback intensity: some of them need intense vibration in order to detect the feedback, while others suffer “pins and needles” sensations and thus prefer less intense vibration. For this reason, a vibration pattern editor has been proposed, allows the application developers (or the museum curators) to prepare various solutions to offer to the end users.
The originality of the discussed prototype is on the approach and the specific solution adopted. Indeed, while those who have previously worked on haptic feedback for mobile users have not considered solutions for supporting blind users, vice versa those who worked on mobile support for blind users have not considered haptic feedback in an unobtrusive manner. Relevant research on the use of haptics and non-speech sound for supporting blind and visually impaired has been carried out in [61], but the experiments reported were limited to stationary situations, i.e. conveying graphical information to the blind by means of haptic output solutions.
Differently from previous work, in this section a prototype system that exploits the haptic channel in a mobile configuration for enabling the mobility of the blind has been described, including its development and testing.

2.4 Final remarks

This chapter has introduced the issues of design and implementation of user interfaces for mobile devices. The case study of a mobile museum guide has been tackled, dealing with several aspects: location awareness, multi-device capability, multimodality and accessibility for impaired subjects.

The initial prototype discussed enriches the museum visits through individual or collaborative games. Its main contribution is in the exploitation of multi-device environments, where users can freely move about with their mobile guide and also exploit large screens connected to stationary PCs when they are nearby.
Both the access to museum information and to the games can benefit from multiple devices available as well as from additional services, such as localization of other visitors on the large screen, which is detected through RFID tags. An authoring environment, which allows museum curators to easily customize the contents for

their museum guides, has also been proposed. The multi-device guide has been deployed in two museums in Italy (Marble Museum of Carrara and Natural History Museum of Calci) with the aim of improving the visitors' experience by extending their interaction with exhibits.
From the discussed experience, it is clear that museum visitors are increasingly more varied in their interests and background, and all such aspects need to be considered in the design of digital guides, which should be flexible enough to adapt to the different needs. The work also highlighted the importance of understanding how new information technologies might change, and can possibly improve the way visitors approach museums.
It has been found that multi-device support to opportunistically and easily use the devices at hand can be useful for better accessing information related to the museum and its artworks. Improve social interaction is also useful to help users to acquire greater knowledge regarding the associated content. The combined use of large screens and mobile devices enables heterogeneous solutions. Users can receive information that depends on their interests and location, whilst freely moving around. Large screens can even be exploited for better accessing more detailed information, additional services and to share games and content with other visitors.

The investigation on the case study of the museum guide has dealt with accessibility aspects, i.e. the mechanisms that enable impaired people to access the capabilities of the guide. A wearable solution for supporting blind people in moving among tagged objects of the museum has been widely discussed. The guide solution has been specifically designed and tuned for museum environments, and exploits haptic feedback for facilitating orientation and obstacle avoidance. The final proposal has been obtained after an iterative design process, including various tests with visually-impaired users. The result is a quite complete solution able to take into account user orientation and position, and supports obstacle detection. It is based on the use of vibrotactile feedback in a novel manner, which is unobtrusive for even crowded indoor environments, easy-to-use. In addition, the vibrotactile feedback is less annoying than vocal one when the sequential suggestions are frequent.
User tests regarding the vibrotactile feedback for obstacle avoidance have been carried out. Statistical analyses on the results do not highlight preferences towards a particular feedback mode. Users clearly motivated their different preferences (some found the continuous vibration “annoying” while others felt it as “safer”), thus a customizable haptic output to let the user choose the preferred mode has been introduced.

3. REPRESENTING SMART ENVIRONMENTS ON MOBILE DEVICES

3.1 Introduction

Device diversity and the availability of applications able to exploit it are steadily increasing. While coping with multi-device environments, users tend to assign different roles to different devices according to the capabilities needed, such as computational power and screen size.

A study about why and how people use multiple devices [37] found out that users rely on various techniques for accessing information across devices, though each device is characterized by its own features, limitations suitability for performing some tasks. Indeed, although the computational resources of mobile devices are growing, specific capabilities are recommended for performing some tasks. For example, watching a long video or looking at high resolution pictures are typical examples of actions requiring proper features, such as a suitably large screen. Thus, users do not use all devices in the same way and tend to assign different roles to devices by preference and necessity.

Modern contexts of use and their multi-device technological availability have been exploited by an increasing number of features and services. Pick-and-drop [93] was an early interaction technique to provide support across multiple devices, in particular to interactively exchange data among them. Kozuch and Satyanarayanan [62] presented a solution for application migration, based on the encapsulation of all volatile execution state of a virtual machine. The migration support was unfortunately limited only suitable for desktops and laptops.

A system for graphical user interface distribution over a federation or group of devices was presented in [69]. The authors focused on desktop and traditional mobile systems.

Migration of Web applications across heterogeneous devices was tackled in [6], where logical descriptions and automatic user interface generation are considered.

In general, researchers and developers are introducing techniques and tools to manage information and activities across many devices.

In this context one important problem is how to support mobile users in discovering and selecting the target device in which to continue their interactive session. This is particularly true when multiple devices are available in the environment, as there is potentially no limit to the type and number of target devices that may be deployed. Enabling the user to easily discover and select the best candidate is problematic especially when s/he has never visited the environment before and does not know what devices are available, where they are located, and their availability for cross-device activities.

A relevant aspect in this respect is to enhance smart environments with automatic detection of events related to user interaction with the current context, such as changes in orientation (i.e. user direction) or proximity of the user with respect to a certain device. Indeed, managing such events can be useful for the automatic triggering of specific functions (e.g., launching a particular service on a target device as soon as it is in front of the user).

In the following sections, an environment to help the mobile user in discovering the devices available in the visited environment is presented. The environment facilitates the selection of target devices in order to enable inter-device operations. The central module of the environment is graphical component that exploits positioning support, based on active RFID (Radio Frequency IDentification) technology, and an orientation capability enabled by a wearable electronic compass. The component, which is called Device Selection Map, provides end-users with graphical feedback about their location with respect to the available target devices by means of iconic representation. The design of the solution is described and two possible representations of the interactive embedded component are discussed according also to the results of a user study.

The combination of the two above-cited relatively well-known techniques (RFID and compass) is also tackled, in order to investigate how it may support the mobile user in realizing where the available target devices are located in an interactive map. Benefits, limitations and possible improvements of the proposed solution are discussed as well.

Location-based services have also been explored by means of supporting ontologies, which is reported in [104]. The investigation proposed in this dissertation, however, focuses on a different issue: how to discover and represent dynamic set of devices that can be selectable by the user. Hence, the study deals with techniques to represent user location information with respect to the available devices in a smart environment. The aim is to provide support for selecting a target device, since many applications could benefit from this technique. An example of location-based support, though limited to exhibition environments, was presented in [31], which described one of the first indoor mobile guides for museums. This type of application, as discussed in Chapter 2, could also exploit other interaction platforms, such as large public screens. Thus, it would benefit from a tool allowing users to get graphical representations of the dynamically discovered devices. Indeed, the users would receive a representation of the target devices depending on their position, that facilitated the selection of the best one in order to continue receiving additional information regarding the artworks encountered.

An interesting approach to the user positioning support was proposed in InfoRadar [18]: the authors enhanced a location-based messaging system with a radar-view of the places in the environment with associated virtual content. The radar-like presentation displays the user in the center of the map and the environment items around her, according to the distance and orientation. The evaluation revealed that some users found the radar-like presentation quite disorientating due to the continuous relocation of the virtual items on the display. As in InfoRadar, the visualization support discussed in this dissertation also exploits localization and user direction detection in real time. However, while InfoRadar provides only a radar-like view of the environment (i.e. user-centered), this section goes deeper in investigating the positioning display strategies to improve their usability.

The study illustrated in [55] also deals with the use of maps for guiding users through an environment. The authors present a mobile guide for fair exhibitions and conclude that the egocentric modality for presenting the map (i.e. user-centered mode) provides the best support for navigation. A user test was conducted involving random fair visitors but, as the authors state, no electronic compass was used for automatically updating the map orientation. The system

evaluated in this section is instead full automated and includes an electronic compass.
In general, representing off-screen objects has stimulated a good deal of research interest. Halo [9] conveys direction and distance by means of rings (named halos) that are centered on the interest points. Since the points are off-screen, only a part of the ring is displayed, and the ring range is proportional to the point distance. Disorder occurs when halos overlap. Burigat et al. [135] compared Halo to scaled and stretched arrows that encoded distance as size and length of arrows, respectively. They reproduced Halo and found out that it improved performance when precise distance was required, and that scaled and stretched arrows were faster and more accurate than Halo in an off-screen target ordering task. Another technique is Wedge [8], which represents each off-screen locations using acute isosceles triangles: the tip coincides with the off-screen location, and the two corners are located onscreen. Wedge showed significant accuracy advantages over the Halo. However, in general such techniques address slightly different types of issues and applications. They tend to provide support for applications representing information such as geographical maps with associated points of interest (such as shops, hotels).
The above mentioned studies are devoted to large scale localization, while this section of the dissertation focuses on smaller areas location awareness, such as rooms populated with various types of interactive devices (PCs, laptops, printers). Thus, in this domain it is important to provide a complete view of what is available (deployed and active) and its features rather than to have indications that there is something of potential interest somewhere off the screen.
Seager and Fraser [94] describe a study on the usability of automatic, manual and physical map rotation compared with static north-up map for pedestrian navigation supported by handheld device. The participants of the user study preferred the physical map rotation (i.e. rotating the handheld to align the represented map with the real world).
An early evaluation on north up map and heading-up map on automotive navigation was presented in [36]. The authors did not find any particular difference, but reported that the rotation angle for the heading-up display must be carefully chosen in order to not affect user's performance. In [54], where egocentric maps on mobile devices are considered, the authors indicate that the users preferred the track-up (or egocentric) view. The authors, after testing the automatically rotated, the physically rotated and the north-up map, conclude that the north-up is the less usable modality since it determines the highest number of navigation mistakes. The use of three-dimensional maps together with textual and spoken instructions was investigated in [63], which reported that different contexts are characterized by different cognitive load: route instructions may be presented according to the context in order to reach a target point.
While [36], [54] and [63] focus on outdoor support for car or pedestrian navigation, this study refers to indoors and involves different issues arising from the different technology used for location and orientation detection. Indeed, the map representation described in this chapter aims to enable the selection of a target device. This task requires users to create a mapping between the device representations in the map and the actual devices in the surrounding physical environment.

A design framework for multi-display reaching techniques has been discussed in [76]. The paper addresses the topic of a user in a fixed position moving objects between a laptop and an interactive table. Thus, the topic is different from the one tackled in this dissertation, which is novel since it considers mobile users (rather than stationary users coping with specific hardware configuration). Nevertheless, if the tool presented in this section were framed within the design space in [76], it would be considered as coupling virtual and physical space. The nature of the destination space is discrete, and in terms of range a room is considered.
The authors of [44] consider the RELATE system and the interaction with co-located devices that a user might encounter in the surrounding environment. They present an experimental comparison of two graphical interfaces (spatial list and iconic map) and one non-graphical interface (alphabetic list) for the discovery and selection of nearby devices. The results highlight that users prefer the iconic map to the spatial and alphabetic list. The authors observe that, due to the limited number of devices used during the evaluation, the scalability of the solution cannot be assessed. They also state that, when the number of devices per room increases, additional visualization techniques are necessary to avoid compromising usability. In contrast to the RELATE paradigm, aimed to support spontaneous meetings among various people (i.e. sedentary situations), the study in this section is about mobile users in multi-device environments who want to select one of the available devices for some reason. Thus, involved users are interested in obtaining a representation of the environment and the correspondent position of the available devices rather than in identifying their relative position. Indeed, it is assumed that the visited smart environment is already configured (i.e. an area map with the position of the stationary devices is defined and stored). By counting on a ready-to-use map, it is not necessary to determine the relative position of the target devices, but only that of the user. A similar approach simplifies the localization infrastructure and prevents to equip each target device with specialized equipment (a sensor dongle or RELATE-like hardware). In addition, RELATE hardware is limited to line-of-sight between devices and has been optimized for co-planar devices. For this reason, instead of RELATE dongles, off-the-shelf technology has been adopted for the system described in this section: RFID (Radio Frequency Identification) active tags are exploited for detecting when the user is in proximity of certain points and a wearable electronic compass for gathering her current direction. The Device Selection support discussed, gathers dynamic information about the target devices through a discovery protocol, which is merged with the users' location, in order to provide them with representations showing the state of the devices as well (e.g., active/inactive, etc.). More detail on the device discovery protocol is provided in the following.
This chapter provides details of a novel technique for supporting device selection through dynamic graphical representations of user's orientation and position, in relation to the available target devices of the visited environment. The design, implementation and evaluation of two possible location-aware representations of the user and the target devices are discussed.

3.2 User-centered and map-centered views

Access to applications across multiple devices is being supported by techniques and tools that researches and developers have introduced. However, in multi-device contexts, target device selection is often problematic, especially in unfamiliar environments, and is still an open problem. One of the main challenges is to provide an effective representation of the smart environment around.
The several representation techniques that have been exploited in the past in different contexts, can be classified into two main categories: user-centered and map-centered. The former is also known as "radar-like" visualization, since it depicts the user at the center of the map as it occurs in avionic and marine radars (a possible application is discussed in [18]) or in automotive navigators. The latter, often referred as "north-up", consists on displaying the user/device (e.g. through an icon) according to his position/direction over a fixed map.
Although the two modalities are aimed to convey the same information (i.e. the relative position/direction of the user with respect of the map is), the different techniques that they exploit may lead to diverse user perception and experience. The following sections in this chapter provide a discussion about such techniques, and draw some usability remarks.

3.3 Application and user study

3.3.1. Design and Main Features

The fundamental requirement for the Device Selection support was to provide the mobile user with a representation of the set of target devices available in the visited smart environment. The display had not to be limited to a list of devices, and had to suggest in some informative way the location of the devices as well. For this purpose, a map view representation seemed a reasonable option. It is known that a map-based user interface typically introduces some bias, due to the simplification and potential small inaccuracies with which it reflects the real environment. Nevertheless, a map can provide more information than a simple device list, a solution often used to indicate only the available devices. Indeed, a map view can show the set of devices according to their type (e.g., using different icons for different types) and, at the same time, giving indications about the colocation of the device(s) and the user. A previous study on indoor navigation carried out in a shopping mall [91] lead to some recommendations for user interface design, such as heavy pruning of details and simplification of the graphical layout. The map representations of the study tackled in this dissertation are consistent indications proposed in previous work. Indeed, in the simplified maps there are icons for each device that, when selected, trigger the visualization of various pieces of related information (device type, operating system, screen resolution, name, IP address, picture of the actual device, etc.).
In general, the smart environment visited by a mobile user could be mainly presented in two ways: with the map fixed and the user icon moving around and rotating, or with the user fixed and the map rotated/translated according to user's movements and direction. In particular, the display strategies discussed in the two following subsections have been adopted.

3.3.1.1 Map-centered approach

In this modality, often referred to as *north-up* or *exocentric*, the map of the environment has fixed position and orientation. The user icon is instead translated and rotated according to her movements. As soon as the user changes location/direction, the position and orientation of her icon on the map are updated accordingly (see Figure 20, left).

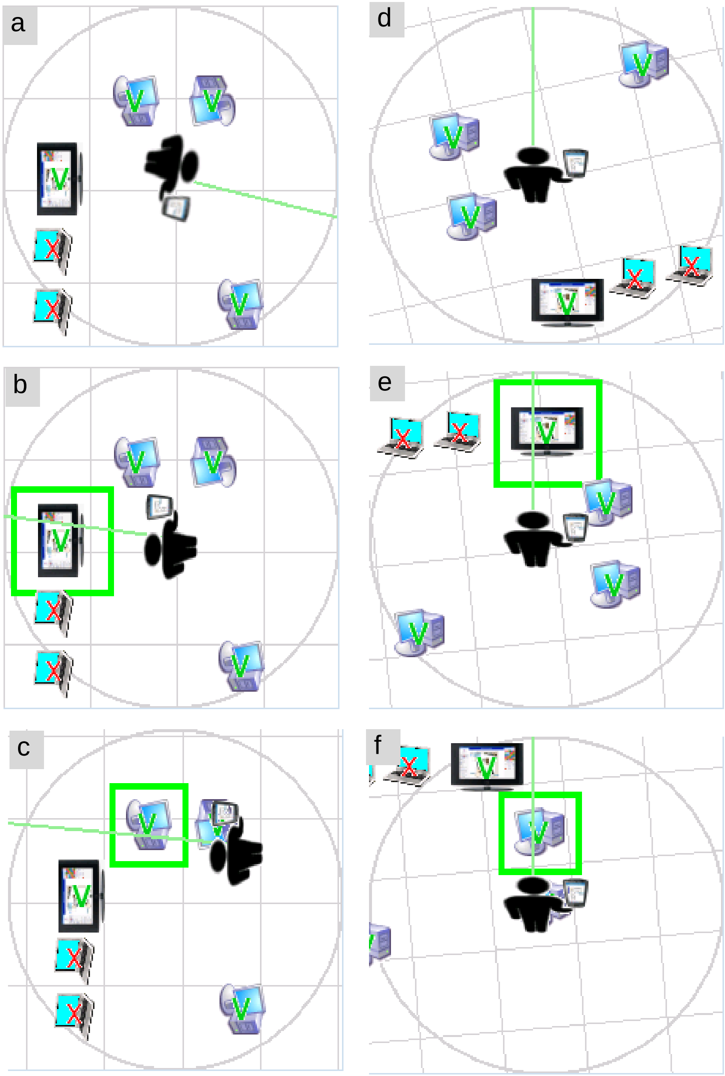

Figure 20. Examples of visualization in map-centred mode (a, b, c) and in user-centered mode (d, e, f): the user in the center of the room (a, d); the user in the center of the room looking at a large screen device (b, e); the user after moving towards a desktop device and looking another desktop device (c, f).

3.3.1.2 User-centered approach

This strategy, often referred to as *heading-up* or *egocentric*, consists of centering the user icon on the graphical container and never changing its position or orientation (see Figure 20, right). As the user moves and/or changes direction, the map is translated and rotated around her.
For example, if the user turns left 30°, then the map is rotated 30° right. When the user is approaching a new tagged device, the device position becomes the new rotation center of the map.
The first version of the tool was evaluated through a first formative usability test to investigate which visualization modality (map-centered or user-centered) was preferred by the users. From the test, that involved 11 users, a slight preference towards the map-centered mode emerged.
Taking into account the observations and suggestions reported during the preliminary evaluation, the Device Selection Map component was enhanced with some functionalities before setting up a further user test. One of the new functions consisted of the icon orientation, which shows the devices' icon oriented by reflecting the orientation of the real device. Another function was the zoom in/out over the visited point. The grid of the map was also introduced for giving an indication about the current zoom factor (this feature could be switched off by the user).

3.3.2. Case study: support for migratory user Interfaces

A possible application for the Device Selection component is the support for migration of user interface. The migration process implies that the user interface dynamically moves from one device to another, with the aim to "follow" the user and to allow her continuing her activities. Thus, when migrating to another device, users can immediately continue from the point they left off since they find all the information previously entered in the source device.
On the source device side, migration is supported by a client software (such as the Device Selection Map), which is a tiny application running in background. The migration client allows to trigger the user interface migration from the current device to another (e.g., from the mobile to a stationary desktop system) and to select the target device. Examples of software architectures supporting migration are presented in [84] and [51]. Among the many applications that can benefit from this type of infrastructure, there are the ones that require long sessions, such as interactive games. For example, a user playing with her PDA outside, might want to migrate the game to a specific desktop system upon reaching her house. However, such solutions for supporting migration either are not able to dynamically discover and represent the available devices [51] or provide limited solutions for this purpose [84].
The developed Device Selection component has been integrated with a migration infrastructure, in such a way that when the mobile user enters a tagged smart area (e.g., a certain room or public environment) s/he is aware of the available target devices for migration. The awareness is provided by the visualization of the environment map with respect to the available devices. The user can thus select a candidate target device and trigger the migration of the application. The migration infrastructure resumes the application on the selected target device. Figure 21 shows the user interface of a simple client with list-based presentation of the

discovered device names as used in [84] (left), and an enhanced client exploiting the Device Selection Map (right). It is worth noting that both the list and the map refer to the same situation, but represent it in different ways. The list-based presentation reports only the 4 devices detected as active. Such a textual list of the names of the available devices is often unusable because users might not know the name/device associations. The Device Selection Map provides instead a more intuitive and informative representation of the deployed devices, including their type, the active state (the icon is annotated with a green "V" or a red "X" indicating whether the device is active or not), and their location with respect to the user. In the example of Figure 21 six icons are shown, but only 4 devices are active: the three desktops and the large screen, which are checked with a green "V" (the two laptops are instead annotated with a red "X").

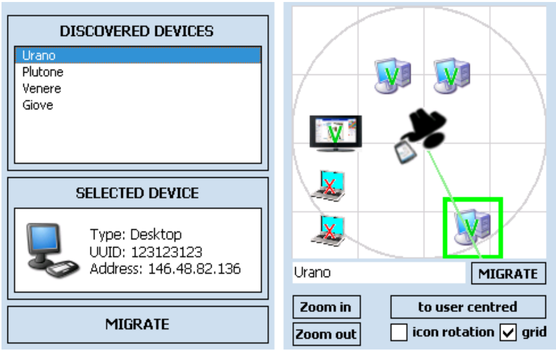

Figure 21. The user interfaces of two migration clients: the list-based (left) and the Device Selection Map-enhanced (right).

3.3.2.1 Enabling hardware

The localization solution for the Device Selection Tool is based on RFID technology and an electronic compass, allowing to detect user direction and position with respect to the devices deployed on the environment.

The enabling hardware is indicated by the lowest layer of the diagram in Figure 22.

The electronic compass is a wearable device that detects user direction with respect to North. The compass peripheral device has been developed in order to satisfy the requirements of this and other applications requiring location awareness (see section 2.3). Since wearability and compatibility with PDAs were needed, the electronic compass has been made battery-operated and Bluetooth-enabled. The current electronic compass is the result of many experimental trials that regarded both user comfort and performance. In its current version it is wearable on a belt and thus it detects the direction of the user's body.

Figure 19 (Chapter 2) shows the detector device that was developed and used for the experiments (it embeds both a compass sensor and distance sensor).

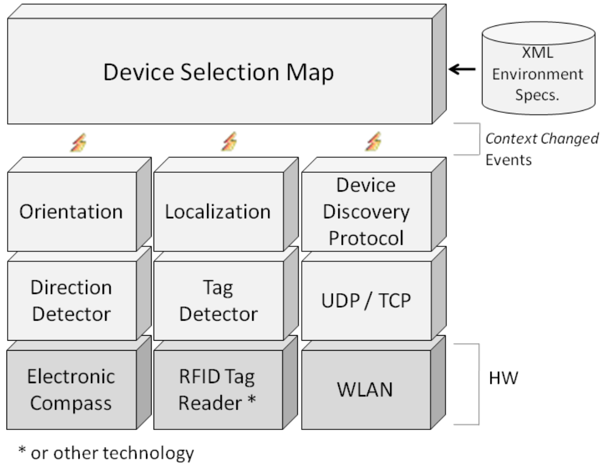

Figure 22. The architecture of the Device Selection environment.

In the near future, many PDAs are likely to be equipped with support for direction sensing and widely available on the market. Thus, the complexity of the hardware may reduce in the future, leading to a simpler and lighter solution.
The RFID tag reader is plugged on the PDA through a Compact Flash (CF) card. It has an antenna of 6 centimeters, which is able to detect a RFID tag up to 5 meters.

3.3.2.2 Software Architecture and Graphical Interface

The first layer above the HW consists of the framework for easily accessing the enabling technologies (i.e. to integrate them into any application) and is independent of the application. *Direction detector* and *Tag detector* are software modules that interface to the electronic compass (to get user direction) and to the RFID reader (to compute the nearest visible tag), respectively. The *Device Discovery Protocol* is a set of functions for dynamically detecting the active target devices using UDP (User Datagram Protocol). The protocol, executed by every device that runs the device discovery client, announces the presence of the local device by "hello" messages and collects the announcements of the remote devices. The "hello" messages contain a short description of the device: name, type, internal and external IP address, screen size and operating system. Such information is stored in the description file of each device. Additional parameters could also be added. However, care must be taken while coping with dynamic information, such as the expected time to perform a migration: a similar information may not be indicated, because it is strictly related to the structure of the application to be migrated (e.g.: a Web page), that is not known before the adaptation module has performed its computations.
In practice, every device joins the discovery multicast group, listens continuously to incoming UDP "hello" messages from other devices and periodically multicasts its own "hello". When a device is about to be deactivated, the discovery protocol multicasts a "bye" message. The list of "discovered" devices contains only those devices that have announced themselves recently (the time-to-live interval is settable in the configuration file), in order to avoid inconsistencies. This is because

when a device is switched off abruptly, without correctly multicasting a “bye” message, (or when the “bye” UDP message is lost), that device is kept in the discovery list of the other devices. One characteristic of the UDP multicasting is that it is implicitly limited to the local network (the gateways typically block multicasting to avoid flooding). This is well-suited to the concept of user interface migration, which involves devices within a well-defined setting.
The third layer from the bottom belongs to the proposed tool, such a layer triggers events when relevant changes occur in the context. The events are then caught by the Device Selection Map, which updates the graphics. For example, a Direction Changed event is raised by the Orientation module only when the user changes direction by a value above the threshold, to lower graphical flickering. The Tag detector triggers a Location Changed event only when a tag signal is detected with a certain strength (i.e. when the user is in close proximity to it). The XML Environment Specifications define the set of areas and devices deployed in. An area may represent a room or a part of it and is basically defined by size, orientation with respect to North and set of items. The main attributes of an item are: type, position within the area, orientation with respect to the area and (if the object is tagged) id of the tag (e.g., RFID). Note that an area item could be a target device as well as an element of the environment that may help the user to orientate herself (e.g., a coffee machine).
A graphical editor for the smart environment allows the creation and update of the area maps: devices can be instantiated and deployed on the map in order to reflect their real position within the room. The editor is a desktop application and produces an XML specification of the environment layout. The XML descriptions can be manually copied onto the devices or downloaded automatically from the server by the device selection client when it is started.
The set of devices deployed in the visited area is compared at run-time with the list of active devices in order to define the ones that are currently available for selection. Thus, the representation of all the deployed devices reflects the real current state, which users see on the PDA. The icon style (annotated with a green “V” or a red “X”) indicates whether each device is active or not.

3.3.2.2.1. Location/orientation detection and devices display modes

Locating the user and sensing her direction is fundamental to inform her about the devices deployed around. The prototype in its current state relies on pre-defined maps of the environment, that can be stored on the mobile device or downloaded on-the-fly from a gateway of the smart environment. The position changes are detected when the user approaches a RFID tag associated to a device, while direction is continuously sensed by the compass and updated in real time on the map.
Localization is event-based: when the Tag detector determines that the user has approached a new tagged point it raises a Location Changed event with the tag id of the new location. Although only RFID tag detection is enabled so far, the Tag detector component can host additional specific threads for exploiting different localization technologies.
The compass detect user orientation in the mobile support. Mapping the current user direction on the graphical representation (i.e.: updating the area map) is simple since the orientation of the visited area is defined at configuration time: the

area orientation is fixed. This aspect is important for enabling the right orientation of the map in user-centered mode and the right orientation of the user icon in map-centered mode. The environment map should reflect the actual layout with respect to the current location and direction in order to facilitate user's orientation.

3.3.2.2.2. Device discovery

The location/direction aids facilitate associating the virtual items to the corresponding ones in the real world (and vice versa) through a graphic overview of the currently visited area. This is done by matching user current position/direction with the static data of the target devices. Dynamic data about the devices is sensed through a device discovery protocol, similar to the UPnP one, executed by the application that supports the device selection capability. The device discovery indicates which devices are actually active among those represented in the static map.

3.3.2.2.3. Management of context-related events

The events that the Device Selection Map is able to manage are: location changed, faced device(s) changed and discovered devices list changed.

The Device Selection Map, as already said, rather than as a standalone component, has been designed to ease device selection in a smart environment by being integrated into a wider application. This can be the case of a mobile client application for accessing functionalities deployed on stationary devices. The orientation capabilities of the Device Selection Map can be exploited even outside of the component to facilitate the identification of the right target device. As an example, a technique has been implemented for alerting the user about the "pointed" device by showing a big font sized message such as "you are pointing me" on the display of that particular device. The benefit for the user is to be sure about the correspondence between the virtual item and the real device (e.g., a desktop PC) and to avoid ambiguity. This is especially true when one or more devices in the room are very close to each other. The prototype client implemented hosts the Device Selection Map and exploits web services deployed on the target stationary devices to provide feedback upon verification of relevant events, such as graphical alerts on the display of the watched device. For example, when the user moves towards a device, the mobile client application detects the location changed event or the new device(s) watched event of the Device Selection Map and calls the web service of the selected device, which shows a text such as "you have selected me" or "you are watching me" in a popup window.

3.3.3. Evaluation

A user test has been performed in order to evaluate the Device Selection Map component enhanced with icon rotation and zoom capabilities.

3.3.3.1 Participants

The 14 participants were all recruited among the personnel of a research institute and were between 24 and 42 year old (M = 31, SD = 5.6). Six of them had previously used orientation supports (such as car navigation systems). They also

rated on a scale from 1 to 5 (the lowest and highest value, respectively) their experience on the use of PCs (M = 4.6, SD = 0.5) and PDAs (M = 3.2, SD = 1.3).
All the participants, at the time of the test, were working in an information technology institute. It is reasonable to believe that such a sample represents well the typical users of the tested support, who would be people with a good knowledge of technology and the various interactive devices available.

3.3.3.2 Environment and tasks

The test took between 10 and 15 minutes per participant and was performed in two rooms. Each user was introduced to the capabilities of the system and instructed about the tasks to perform. In the first room the user could freely move around, taking some familiarity with the position/direction feedback and trying the various options (zoom, icon rotation, switch to user-centered mode). At this stage, the user was also encouraged to ask any question to the test supervisor regarding how the system works.
In the second room, which was about 4 meters wide and 8 meters deep, 7 devices were deployed: two laptops, three desktops, one digital TV with large screen and one printer. Just four devices were RFID tagged, the system was tuned to detect the tag from up to one meter and the participants were informed about that. Figure 23 shows an overview of the room. The relevant devices are annotated as: 1-2 (laptops), 3 (digital TV with large screen), 4 (printer), 5-6-7 (desktops).

Figure 23. Overview of the room where the user test took place. The relevant devices are annotated with 1 to 7 numbers.

Each participant had to carry out specific device selections by pointing exactly the requested device with the body and clicking the button “migrate”. The device to select was indicated through a descriptive sentence (e.g., “the desktop near the window” or “the laptop next to the door”) by the test supervisor. For feedback purposes, the description of the device currently “watched” by the user was automatically displayed by the application on a label in the GUI. Users had to select six devices: three in one type of representation and three in the other. For

avoiding biases due to the learning effect, half users started with the map-centered mode and the others started with the user-centered mode.
During the test session, each user was free to use any additional capability of the Device Selection Map, such as zooming, icons rotation, grid view.

3.3.3.3 Results

The time elapsed between the supervisor's request and the right selection was logged: in map-centered mode, it varied between 2 and 44 seconds (M=11.9, SD=9.9); in user-centered mode, it varied between 2 and 150 seconds (M=38.8, SD=32.3). As a consequence of the high diversity of the time needed by the users for selecting the target devices in user-centered mode, the Standard Deviation is also high. A Wilcoxon test (N=14, Z=-3.297, p=0.001) on the time logging revealed "high significance".
After the test sessions each user answered to a questionnaire providing some informal observations/suggestions and an overall rating on the solution. The two presentation versions were rated on a scale from 1 to 5 (the most negative and positive value, respectively). On average, the user-centered approach was rated 3.3 (SD=0.9) and the map-centered 3.8 (SD=1.2). Eleven users preferred the map-centered and 3 the user-centered. The results are also summarized in Table 1.

	Map-centred	**User-centred**
Mean time for selection / Standard Deviation (seconds)	11.9 / 9.9	38.8 / 32.3
Preference by users	11	3
Average rating / Standard deviation	3.8 / 1.2	3.3 / 0.9
Difficulties encountered	None 10	None 4
	Some 4	Some 9
	Many 0	Many 1

Table 2. Summary of the test results.

Most users reported that the map-centered approach is more intuitive and that is easier to find the target devices due to the static representation of the environment that provides an overview faster to analyze at different times. Someone also found that it is more realistic because it shows the user icon approaching the devices, differently from the user-centered in which the map moves while the user does not. Among those who preferred the map-centered, a user said that its benefits there is the better fluidity of the graphical refresh when the direction changes.
The participants who preferred the user-centered strategy found it intuitive to see the devices repositioned on the component by reflecting the current vision of the environment (i.e. the desktop in front of the user is drawn on the upper area of the component) making it possible to observe just one portion of the component to see, for example, what is in front of him/her.

Even if all the participants received the same instructions before starting the trial, it was observed that some of them, after making little moves (e.g., a half meter), looked at the PDA waiting for the upload of their position on the Device Selection Map. They simply had forgotten that the position update occurs when particular points (i.e. a RFID tagged point) is reached. Some users tried to point the devices by rotating the PDA towards them. This misunderstanding was cause of troubles because the system detected the direction of user's body and not that of the PDA (since the compass was attached to the user's belt).
In general, the comments of the users were positive. Indeed, the question asking whether the approach was useful to support user orientation and ease device selection received an average evaluation of 3.8 (SD=0.8). Another question was about whether particular problems were encountered in using the support (possible answers: none/some/many): for the map-centered 4 declared to have found some difficulties and the other 10 none, while for the user-centered; one found many difficulties, nine some, and four none. Also this result shows that the map-centered was found easier to use.
Regarding the use of the icons rotation mode, 8 users out of 14 judged it positively. The others told that rotating the icons in two dimension is not realistic, and thus it does not provide any useful indication to the non-experts.

3.3.3.4 Discussion

The preference towards the map-centered modality, that clearly emerges from the reported results, is related to the requirements, the task supported, and the capabilities and enabling technologies of the Device Selection tool. In general, the indoor localization is still an open problem, especially when coping with very short-range navigation. Thus, a possible way to improve the usability of the solution, alternatively to upgrade the hardware, might be to try to optimize its performance.
More than one participant agreed on the need to improve the precision of the localization, but they also agreed on the usefulness of some feedback on the change of position. An example could be an alerting sound followed by a short animation showing the user icon moving towards her current position. It is interesting to note that several users suggested adopting some of the paradigms used by well-known navigation applications (e.g. Google Maps), such as the indication of the current zoom status by means of a numeric value or a slide-bar. Other indications suggested mapping the main elements of the rooms, such as doors and windows, besides the target devices.
As stated by the authors of [94], it can be assumed that the difficulty of recognizing a map that rotates without looking it could be an important cause of issues when coping with wide maps (i.e. with representations more complex than simple turn-based instructions). Decreasing the flickering could be useful to increase the usability of the user-centered modality, even if it was not the main concern. This can be obtained by using more vector graphics instead of oriented bitmap/icons since a lower number of pixels would be repainted for each direction/position change.
A complementary feature could be the animated re-orientation of the map on request: the map would rotate only when the user asks for it. Since the application is able to detect orientation changes in real time, it may be also able to compute the animation in background, in order to play it when needed. An interesting

advantage of this strategy is the efficiency, especially for devices with limited capabilities, because the elaboration can be performed between the change of direction and the request of map re-orientation.
The outcomes of this study might not be put in strict relation with those regarding the modalities of automotive or pedestrian navigation (i.e. heading-up/north-up) because of the differences both in the relative precision of the localization systems and in the features of the tasks, as well as in the context of use. Before performing the user test session, a preference towards the user-centered mode was actually expected, since that is the same modality used by in-vehicle and pedestrian navigation systems. During the test it was clear that most users preferred the map-centered even if most of them were familiar with generic navigation systems. A reasonable motivation can lie in the different structure between the road navigation and the indoor environments. Indeed, outdoor environments tend to be more structured than indoor ones because they contain elements that are easy to be mentally matched to the map objects. Thus, intuitively, a road along which the user is walking/driving, especially if it contains an intersection, is easier to match with its map representation than a desktop device of a room with its icon in the Device Selection Map (and this is particularly true when the environment is populated by many devices).
The tested maps are instead defined by a set of items in the environment and, when they often change position in the map, it can be difficult then to associate them with their correspondences in the real environment.
The authors of [54] discuss about the importance of environment structure in aligning the map with the real environment (especially in manual alignment).
Users' performance and the preference towards the user-centered modality in the Device Selection tool could increase with more structured maps.

3.4 Final remarks

This chapter has been devoted to describe the design and implementation of an interactive tool for mobile devices, aimed at facilitating target device selection in smart environments, which can be useful for example in supporting migratory user interfaces. The user study conducted has highlighted a significant preference for one specific representation modality: the map-centered display. This preference is likely to be motivated by the type of task the participants had to accomplish (mapping device icons in the map with real devices in the room) and the indoor technology.
Considering the appreciation the users expressed for the display indicating the orientation of real devices on their icons, further work is desirable on this aspect.
In particular, it could be interesting to manage more context specific events related to the orientation of the devices (e.g., to trigger a "device pointed" event only when the user is facing towards the front side of the device, and not whatever side as happens now).
In addition, information related to the state of the selectable target devices, such as time-related features, might be also integrated (e.g., the expected time before a target will be available).

4. MOBILE DEVICES AND USER INTERFACE MIGRATION

4.1 Introduction

People are ever more exposed to ubiquitous environments, which are characterized by the availability of several interactive devices with diverse interaction resources. From the exploitation of such resources, invaluable benefits for an effective user experience would arise.

In this context, interactive migratory user interfaces offer the added value of enabling users to migrate across various types of devices (e.g., moving from stationary to mobile devices) while preserving the task continuity.

Migration implies the possibility for the user to select a target device and to automatically activate on it a suitable version of the user interface running on the source device. The newly activated version must be adapted to the target device features, and must have the same state as on the source device. The state of a user interface includes the result of all the previously performed interactions, such as (but not limited to) the values entered or selected by the user.

Various modalities of migration can be identified, depending on several parameters: the number of source and target devices, whether the entire user interface or only a part of it is involved, etc. In particular, partial migration concerns moving only a portion of the application to another device. An example scenario is the user interacting with a desktop or large screen system, who has then to leave for some reason, and wants to continue the interaction through a mobile device with only a part of the application. This can be due either to application complexity or limitations in the mobile device (for instance, currently most mobile devices do not support Flash applications).

Mashup-like applications, which tend to be particularly complex and made up of various perceivable components, are an example of possible context where partial migration is particularly useful.

Model-based approaches have shown good potential in managing the complexity of multi-device environments (see for example [43, 139]). The use of abstract and concrete logical descriptions (respectively: independent from the target platform, and dependent from the target platform but independent from the implementation language) enable to better support interoperability across various types of devices and/or implementation languages, as discussed later on.

4.2 Requirements and benefits

Nowadays, it is extremely common to see users performing their tasks using various devices ranging from the traditional stationary desktop platform to mobile devices with disparate multimodal interaction resources. While in the past the content that was accessible through desktop systems was substantially different from that accessed and displayed with mobile devices, the progressive enhancements in the capabilities of mobile devices have enabled users to access similar content through different devices, even if their set of sensors (GPS, accelerometers, etc.) still makes them suitable for different sets of tasks. In general, the rich availability of various types of interactive devices has increased users' expectations regarding the provision of suitable mechanisms empowering

them to perform their tasks using opportunistically the devices available in the current context. However, by now, users' expectations have not yet been adequately fulfilled. Often all this technological offering is not exploited as it could be and when users perform cross-device service accesses they encounter various usability issues: poor adaptation to the context of use, lack of coordination among tasks performed through various devices, inadequate support for seamless cross-device task performance.
One potential source of frustration for the users is the inability to continue to perform the tasks when they have to change the interaction device. In such cases, users either have to manually perform some activity in the first device in order to save the up-to-date interaction state and then more easily reconstruct it afterwards on the new device, or, in the worst case, they have to start their activities over again from scratch when moving to the second device.

4.3 Partial migration

Ubiquitous environments suggest innovative uses of existing applications. This section presents techniques and solutions for partial Web interface migration, which let users interactively select parts of existing Web pages on a source device and have them migrate to a target one.
The underlying supporting platform is able to exploit logical user interface descriptions and transformations. This environment is particularly useful for supporting mobile users accessing complex Web applications, such as various emerging mash-ups.
The following discussion deals with a solution supporting user-driven partial migration, its main characteristics, and the architecture of the migration platform supporting it. The approach is illustrated starting with an example of partial migration for a widely known Web application, such as Amazon, in order to show its use and potentialities. After discussion of related work, introduce the approach and the software architecture of the platform supporting partial migration are described.
An example of use of the solution in a widespread Web application is shown, and a the results of a user test are finally reported.

4.3.1. A solution for partial migration

The proposed approach aims to provide a general solution for Web applications implemented using (X)HTML, CSS, and JavaScript. Applications based on languages such as JSP, PHP, ASP are supported as well, because the approach considers one page at a time on the client side. For this purpose, it concerns only those pages that are actually accessed by the user. Another advantage of the solution proposed is that it makes Web applications migratory regardless of the authoring environments used by the developers. It enables the applications to migrate, without requiring the use of any specific tool in the development phase, thus even if the developers never considered migration. This is obtained through the use of an intermediate migration server, which includes reverse engineering techniques that create the logical descriptions of the Web pages accessed on the

fly, which are then adapted for the target device. An implementation with the state of the source version is dynamically generated.
Previous work by the research community has addressed, from different viewpoints, some of the issues raised by partial migration. A discussion on exploiting logical descriptions for this purpose, with example applications, is reported in [7] but without providing a tool able to support end-user driven partial migration in general.
Partial migration can be related, to some extent, to the issues connected with Distributed User Interfaces (DUI). An evaluation comparing various distributed user interface configurations is reported in [68]. In this area, [73] describes a toolkit for deploying distributed graphical UIs. It is based on a widget distributed structure composed of two main parts: one is the "proxy" of the widget and remains stationary within the process that created the widget; the other part, the renderer, is distributed and migratory and the user can interact with it. The toolkit is based on a peer-to-peer architecture in which a multi-purpose proxy is connected to one or more rendering engines able to render (partially or entirely) a graphical user interface. The granularity of distribution is not limited to the user interface groups level but can extend to the widget level and even down to the different components of the widget itself.
In contrast, the solution proposed in this dissertation is client-server like and the distribution extends to the granularity of the single interactor. It is worth pointing out that a deeper granularity was considered unimportant for the goals of the study, and thus has not been achieved. Another fundamental difference is that [139] requires that the user interface be implemented using an extension of the Tcl/Tk toolkit, while the solution proposed in this dissertation is for migration of any Web application developed with the standard Web languages (XHTML, CSS, JavaScript).
The issue of distributing a user interface onto multiple devices is also analyzed in [88], focusing in particular on how to leverage legacy applications to attain the new distributable features easily. The instant messaging (IM)-based architecture is aimed at reducing the additional work needed for extending a user experience to span among multiple personal devices of a certain user. The IM-based architecture presented treats devices as users (giving each an IM account), and affiliates devices to users. The server maintains a list of personal devices for each user in order to determine which messages an application can receive (by default, only those from the user's own devices). Users run a single client on each of their own devices that connects to the server. The various applications connect to that client, which routes the messages to the server. It is worth pointing out that the relations on which this infrastructure is based include a strong limitation that narrows the set of devices that can be interconnected to each other (only the personal devices of a user). Instead, fully migratory applications should be able to opportunistically exploit the devices in the environment (even the devices not owned by the users but accessible to them).
Obje infrastructure [42] supports building interoperable systems without the need to have prior knowledge about them. It defines "meta-interfaces" as sort of agreements on how to achieve compatibility at runtime. Rather than requiring that communication specifics be built in at development time, a set of meta-agreements is used to allow devices on the network to interact each other. The approach presented is called recombinant computing (in the sense that devices and services

can be arbitrarily combined with each other) and shifts some of the knowledge necessary for communication from development-time to runtime. In order to achieve interoperability in this model, three criteria must be met: i) the interfaces supported by devices must be fixed, to ensure future compatibility; ii) the interfaces must be minimal, to facilitate interoperability; iii) such a set of interfaces must be generic. Such fixed agreements become meta-interfaces that specify the ways in which the devices can acquire new behavior to interact with each other. This new behavior takes the form of mobile code that is provided by devices on the network to their peers at the time of interaction. While the motivation of the Obje architecture was to provide an infrastructure for opportunistic interoperation in device-rich environments (as in migration) its approach tends to address problems of interoperation rather than migration.

Huddle [80] is a system that automatically generates task-based interfaces for a household with multiple appliances connected in. Huddle is able to generate interfaces without requiring substantial programming specific to each system of devices, but using instead models of the content flow within the multi-appliance system. The modeling work is divided among the manufacturers of the appliances. Huddle is implemented on top of the Personal Universal Controller (PUC) system introduced in [79], which previously generated remote control interfaces only for individual appliances. Like PUC, Huddle users control the home appliances through mobile devices. From each device, Huddle receives an abstract specification of its functions, which also includes a description of the appliance's physical ports and internal content flows. Huddle does not support migration across devices, which implies starting an interactive session with one and then continuing with another from the point in which the session was left off.

An original solution to support Web session migration using dynamic 2D-barcodes is presented in [2], but without user-driven selection of the interface parts to migrate.

Multimedia adaptation in ubiquitous environments is discussed in [27], which however does not support persistence of the interactive part of a web application across various devices. Collapse-to-zoom [10] is an interactive technique that allows users to interactively compose the interface parts to show on a mobile, but without any migration support.

In general, it is clear that, while a number of approaches have been put forward for the design of multi-device interfaces, including migration (see for example [84]), none of them has shown a general solution able to work on Web application compliant to W3C standards for supporting end-user driven partial migration from desktop to mobile systems.

With the proposed solution for partial interface migration, users access a Web application on a desktop system, interactively select the relevant portions of the interface and migrate them to a different target device.

An example scenario is a user accessing Amazon website and aiming to find some interesting book on usability. The user opens the advanced search page and starts to fill in the various fields of the form in order to narrow the set of possible results (see Figure 24). While editing the request, the user realizes that s/he has to leave in order to catch a train but s/he would like to continue her task on the move.

The Amazon Web page is rather complex and the user does not need all of it for the bibliographic search. Thus, the user activates the migration client, which allows her to select the form and the navigation bar and migrates them to the mobile device. The state of the form (elements selected, text entered, etc.) is preserved, and the user can happily continue the search while moving to the train station.

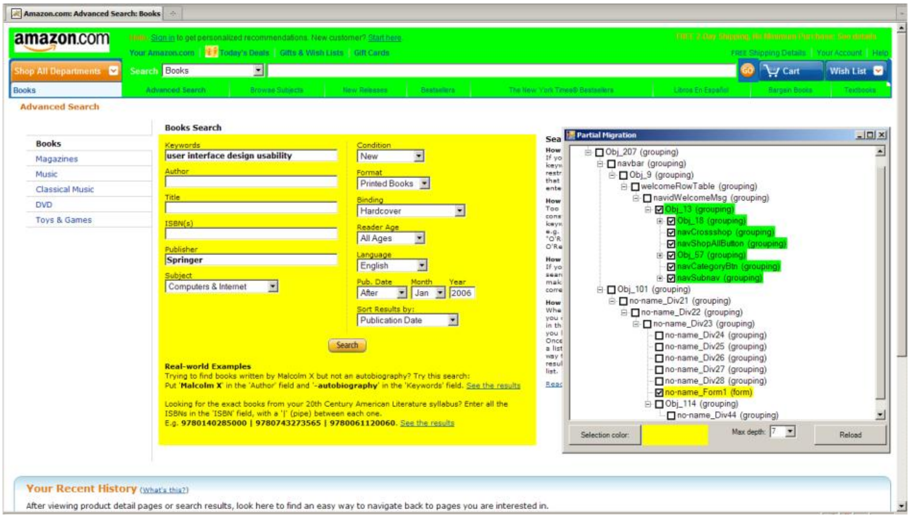

Figure 24: The window that allows the user to select the parts of the interface (in this case Amazon) to migrate (foreground) and the corresponding parts of the interface (background).

In this process of dynamic end-user composition of the interface portions to migrate, the user is supported by a Migration Client (see Figure 24), which displays the hierarchical logical structure of the page currently active in the browser.
The hierarchical representation is obtained by a reverse engineering module [14], which will be described in the next sections. The user interface components displayed are associated with identifiers taken from the HTML+CSS code or automatically generated, if not available in the code. Since sometimes such identifiers are not particularly meaningful, it may be difficult for users to identify what the corresponding user interface parts are. Thus, the migration client has been enhanced with an interactive feature in order to overcome this issue: as the user moves the cursor over the elements representing of the hierarchy, the corresponding elements in the Web user interface are highlighted by a change in their background color. The user can interactively select the color associated with the background of the currently selected part. The user can also perform a multiple selection of the parts to migrate. When an intermediate element is selected in the hierarchical representation of the page structure, its sub-elements become highlighted as well. In the event that the hierarchy has too many levels and becomes difficult to manage, the user can specify the maximum nesting depth to show. In the Amazon search case depicted in Figure 24, the form (highlighted in yellow) and the navigation bar (highlighted in green) have been selected. The color highlights are displayed in the original page are consistent with the ones of its logical structure shown in the migration client.

Figure 25: The panel for selecting the migration target device.

Once the user is satisfied with the parts selected, then s/he chooses the target device and presses the migration trigger button (see Figure 25). The target devices that can be involved in the migration process must run a light software that allows them to dynamically discover each other when active. Thus, every the migration client is always able to show an updated list of available devices. The host names in the list are enhanced by a figure highlighting the type of the corresponding device (desktop, PDA, smartphone, large screen, etc.) as shown in Figure 25.

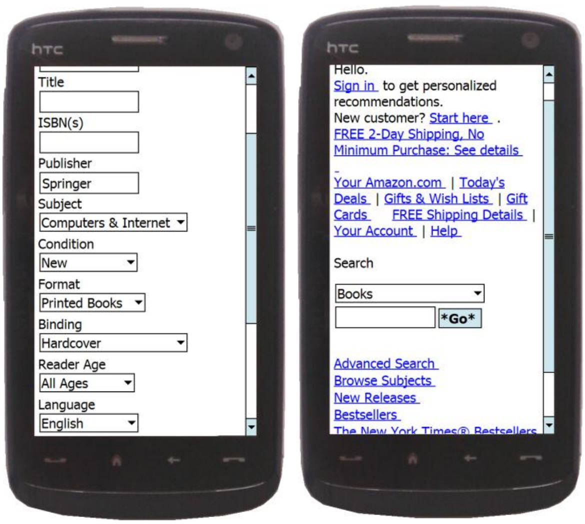

Figure 26: The Amazon web page considered in the example and adapted for the mobile device after migration: the form (left) and the header part (right).

Once the migration is triggered, the interface parts selected are presented in the target mobile device at the point in which the user left off. In the described example, migration took place at editing time, thus the form appears on the mobile device with all the input provided by the user on the previous device (pull-down menu elements selected, text entered in the text fields) as depicted in Figure 26. The parts selected in the described migration process are adapted to the target device. The adaptation, in some cases, includes the splitting of the desktop page into multiple mobile pages too. Figure 26 shows how the splitting is done in the described example: the top part of the original desktop page is accessible on the target device through a different mobile page.

4.3.1.1 Migration platform architecture

The proposed solution exploited logical user interface descriptions. A previously developed XML-based language (MARIA [85]), oriented to support ubiquitous interactive applications, has been used for this purpose. The migration platform dynamically builds the logical structure of a user interface through interactor composition operators, that indicate groups of logically connected elements or relations among such groups (e.g. when a set of controls are associated with a certain form). This subsection describes how this logical structure is made accessible to the users, so that they can interactively select what parts to migrate and to which device.

The current approach assumes that a Web desktop version of the application exists, which does not seem a big limitation given the wide availability of this type of applications. The desktop version provides the basic content for automatically creating the versions adapted for the other platforms (e.g. the mobile one).

The architecture of the proposed solution is summarized in Figure 27.

In the architecture (and in Figure 27 as well) the desktop has been considered as the source device, and a mobile device as the target one. This is because partial migration typically implies a reduction of interaction capabilities when moving from the source device to the target device.

All the devices that can be involved in the migration are detected by the device discovery protocol, which is continuously active to detect changes in device availability. Such devices should be able to run Web applications, therefore they have to be Web browser-enhanced. Each device involved in migration has to run a thin software (the migration client, as previously mentioned), which allows users to know what devices are available, to select the one that should be the migration target, and to trigger the migration. The Migration Client application is implemented in C#. It is worth pointing out that, with respect to Figure 27, the source/target devices included in a single oval are supposed to include the fundamental components for enabling migration (i.e. the Web browser and the Migration Client).

The device discovery support is based on the exchange of information regarding device presence. Communication is enabled by sockets, which can be accessed through various implementation languages. The device discovery notifies (see the Device Discovery phase labeled (1) in Figure 27) the presence of the associated device to a known multicast group. The group consists of all those devices that are running the device discovery software. Such devices are available and can be involved in the migration.

The user interacts through the Web Browser on the source device to request a page (2) from the concerned application server. The request is captured by the migration server (which includes a proxy server), which asks for the application server for the original page (3). The latter provides the migration server with the original requested page (4). The migration server annotates the page and returns it to the Web Browser of the source device (5). It is worth pointing out that this process is repeated for every page the client device accesses to.

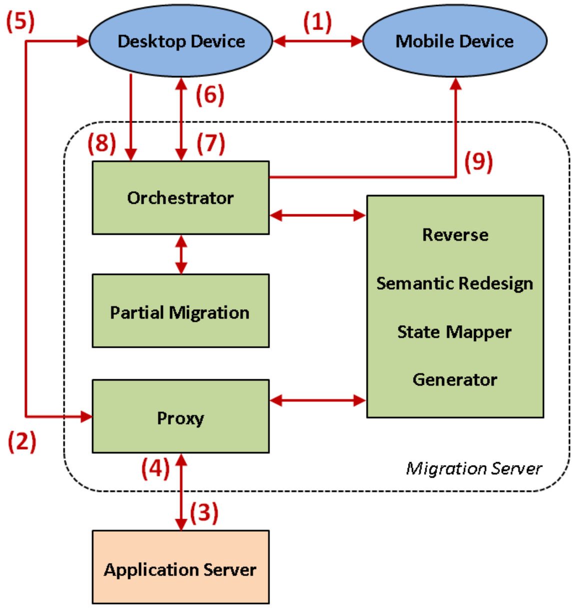

Figure 27: The architecture of the proposed solution.

When the user selects the migration options on the client, the migration server Orchestrator communicates with the Reverse module. This is done in order to call the Reverse module functionalities for the analysis of the page. The Reverse produces the Concrete User Interface (CUI) description associated to the current page and passes it on to the Orchestrator. One specific functionality of the Partial Migration module is to parse the CUI provided by the Orchestrator. The result is a list of user interface components according to the hierarchical relations among them. Such list, organized according to the hierarchy of the composition operators appearing in the main presentation, is sent to the source device (6), in order to let the migration client display the tree-like view of the page structure. The user is then able to specify which components to migrate by selecting them in the migration options form.

When migration is triggered (7), the selection cardinality is considered. If the user triggers a migration request and a subset of the components is selected on the list, then the migration is considered to be partial. In this case, the Orchestrator requests a subset of the CUI from the Partial Migration module, according to the sub-list of components selected by the user, and forwards it to the Semantic Redesign. The type of analysis and reduction that is performed by the Partial Migration module for obtaining such a subset of the CUI will be described later on.

The state resulting from the user interactions (elements selected, text entered, etc.) is automatically transmitted through an Ajax call to the migration server (8). The

Ajax script is included in the annotations performed by the migration proxy server when the page is loaded. The annotation scripts also capture the Web page state through its DOM and provide it on request.
State information includes the last page element that had the input focus in the source version. Therefore, the version for the target device is generated with the same focus it had on the source device before activating the migration, and is sent to the mobile device (9) allowing the user to continue the interaction precisely from the same point s/he left off. The Reverse Engineering part is able to build corresponding logical descriptions from (X)HTML, CSS and JavaScript implementations. If the Web application contains Flash or Java applets, then the reverse is not able to analyze such code. In this case, the applets are either replaced with alternative content provided by the application developers (such as images) or forwarded to the target device "as they are" if the target browser is able to execute them.

The *Semantic Redesign* module transforms the concrete description (specific for the source platform) to the one that refers to the target platform. In general, concrete descriptions assume the existence of some interaction modalities but are independent of the implementation language, while the abstract descriptions are even independent of the interaction modalities. At the abstract level there are, for example, concepts such as selection, edit, activation. At the concrete level, for a graphical device, the selection object can be refined into a list or a radio-button or a pull-down menu or other similar techniques. Such elements can be implemented in different languages. The abstract and concrete vocabularies contain concepts for structuring the user interface as well, such as grouping and relation. The semantic redesign transformation aims to map elements of the source concrete interface into ones that are more suitable for the interaction resources of the target device. The semantic redesign uses the abstract level to identify the type of interaction to support. Then, suitable concrete refinements are identified for the target platform. Thus, for example, in a desktop-to-mobile transformation the possible target concrete elements are characterized by a more limited usage of screen space while preserving their semantics (i.e. the effect that they have on the interactive application). The objective of the State Mapper is to update the concrete user interface for the target device (and which has been delivered by the semantic redesign module) with latest information regarding the state of the UI contained in the DOM of the page just before migration. After obtaining the new concrete user interface description for the target device (updated with information about the state), the Generator module implements the final user interface in a language supported by the target device considered.

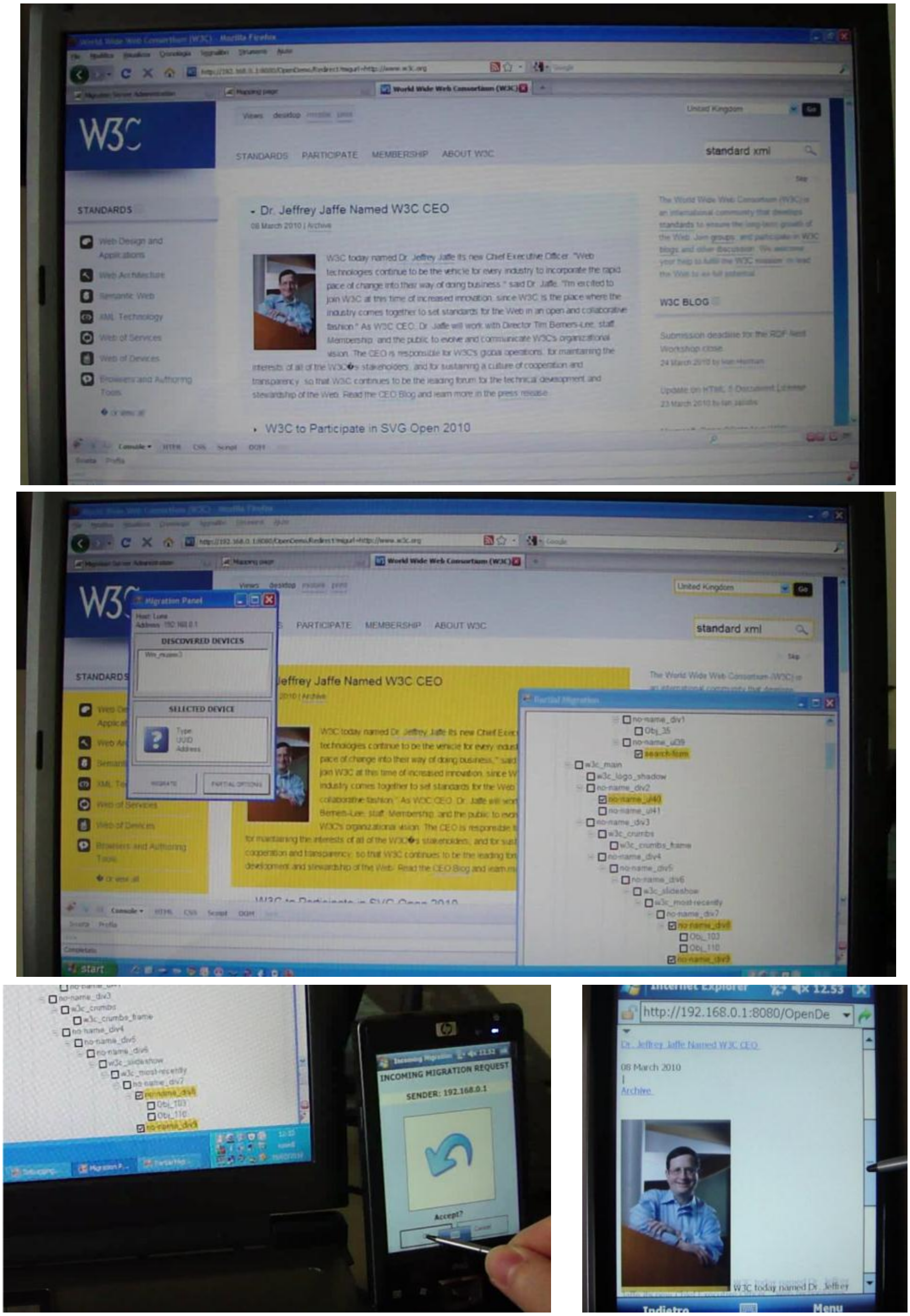

Figure 28: The original web page (top), the migration client and the page highlighting the components selection (center), the incoming migration request on the PDA (bottom left) and a part of the migrated page on the PDA (bottom right).

The *Proxy Server* module plays a role whenever a browser on a client device requires access to Web resource (i.e. page, external CSS, JavaScript, etc.). Indeed, every request to the application server is filtered by this module, which accesses the application server to obtain the original resource and annotates it by including scripts, and changing the links so that next requests will pass through it. The included scripts, in general, manage migration-related issues, such as (but not limited to) capturing the UI state when migration is triggered. The scripts also enable the change of the background color of the selected part(s) for partial migration. This is done through an iterative Ajax script launched at load-time, which is notified, via the proxy server, every time a selection event occurs in the partial migration window of the migration client.
When the user triggers the migration, a dialogue box will appear in the target device in order to allow to check that the correct migration has been performed and confirm it. When the confirmation is sent from the target device, a notification message appears on the source device.

Figure ***28*** shows another example of application of the migration tool on a different Web site (W3C), including such notification message on the target mobile device.

4.3.1.2 Partial migration: preliminary strategy
In this section an example of the type of processing required to obtain partial Web migration is presented. The example considers the Amazon home page. Figure 29 shows the page by identifying the various logical parts with colored rectangles. Such logical parts are identified by the reverse engineering module by analyzing the code and identifying typical tags used for containing purpose (e.g. DIV, FORM, TABLE, etc.).

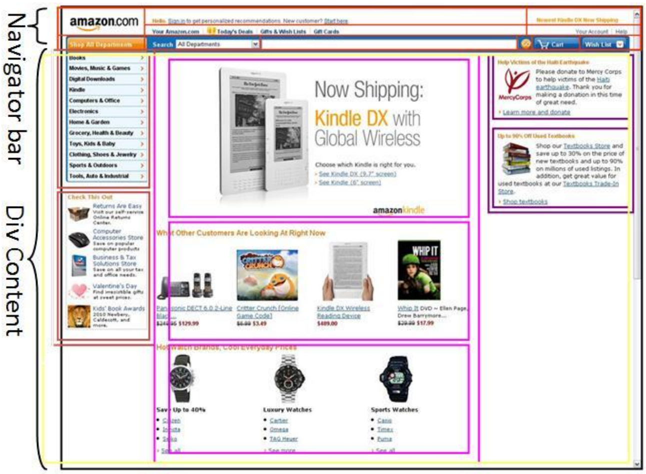

Figure 29: Highlighting the logical structure of the page considered in the example.

The resulting logical structure is shown in Figure 30. It is worth noting that the page structure has a root element, consisting of two elements: the top part (including a navigation bar) implemented through a layout table, and a content part defined by a DIV. The content part is in turn composed of three main blocks of grouped elements, each defined by a DIV: one on the left, one in the center, and one on the right.

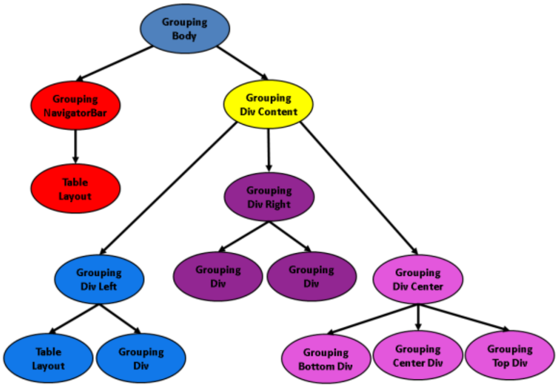

Figure 30: The logical structure of the page considered in the example (at the beginning, before a partial migration).

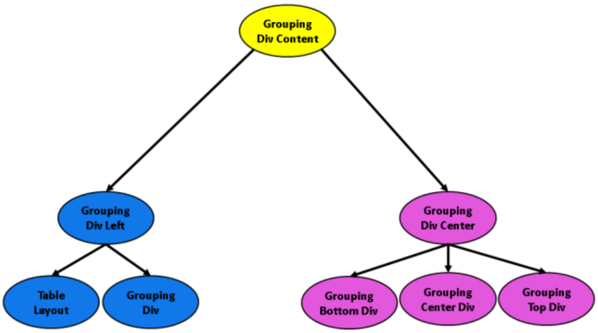

Figure 31: the resulting logical structure of the UI, after the tree pruning and reduction due to a partial migration request.

If the user selects the two parts framed by rectangles in Figure 30, then the structure derived by the Partial migration Module is shown in Figure 31.
The new root of the tree becomes the Grouping DIV Content (the original root in Figure 31) since Body has been replaced by its only “survivor” child.
The underlying algorithm includes three basic steps. First, it prunes all the parts of the tree (i.e. CUI nodes) that are connected with parts of the UI that have not been selected (interactor pruning step). In this way, a first draft of a partial CUI is obtained. The connections to other pages defined by UI elements that no longer belong to the new partial CUI are deleted (connection pruning step).
The final step of tree reduction discards any useless redundancy in the logical structure of the tree. This is because it might occur that some nodes keeping in the

CUI after the pruning step turn out to have only one child. In these cases the algorithm carries out a reduction step, consisting of replacing the nodes that have only one child with the child itself, and applying the same step to the child recursively. Afterwards, all the intermediate nodes have at least two children.
This reduction step is particularly relevant for pages with many nested levels. Indeed, if the structure is quite complex and the user selection has focused on a part of the page that is quite deeply nested, it might happen that chains of many (intermediate) nodes having just one child remain. The reduction step is aimed at deleting such intermediate chains of composition operators that in the original CUI had several children, but that after the pruning step keep with only one child. In any event the semantics of the page is still preserved.
A further advantage of the reduction step is the consequent improvement in the performance of the tree analysis. Indeed, since the useless intermediate nodes are removed from the tree, they will not be analyzed anymore by the other modules (e.g. the Semantic Redesign module, which receives the partial CUI in input for performing adaptation to the target device).

4.3.1.3 User test

The test involved 10 users, with average age of 26.6 year old. All had a university degree and were recruited in the community of a research institute.
The participants received a short (about five minutes) introduction to the migration concepts, and then a list of possible Web pages were proposed. Such pages belonged to widely known Web sites, such as BBC, W3C, Google advanced search, newspapers. The users had to briefly familiarize themselves with the migration client, which allowed them to select parts of the Web pages. Thus, they moved the cursor on top of the logical structure and saw the corresponding elements in the Web page highlighted. Then, they had to select the interface parts they were interested to migrate and the target device. The parts to migrate were selected through the check boxes on the tree nodes of the “Partial Migration” window, while the target device was chosen from among the devices listed on the “Migration Panel” (see Figure 25).
Once the migration occurred they had to analyze its result and continue the Web navigation.
After the test, participants filled in a questionnaire to rank various features of the tool tested, and could provide further informal comments.
The quantitative questions were associated with a Likert scale on a 1 (lowest) to 5 (highest) scale. The obtained results are reported as arithmetic means plus or minus standard deviations.
The users highly rated their knowledge on mobile devices (4,1 ± 0.87). The participants had considerable previous experience in the navigation of Web pages on mobile devices (3.7 ± 0.82) as well. The majority of them had already heard about partial migration of UIs.
The panel for displaying the tree-like logical structure of the UI and allowing the user to select the elements to be migrated was judged quite well (3.9 ± 0.73). The technique used for highlighting the associations between the elements in the hierarchical representation of the logical structure and the respective parts of the UI was rated very highly (4.7 ± 0.48). The users found it easy (4.7 ± 0.67) to retrieve on the mobile device (target device) the information that was originally provided in

the desktop (source device), such as the content of the forms. Moreover, they managed well (4.2 ± 1.3) to continue the interaction on the mobile device, from the point they left off on the desktop.
The least flattering comments regarded the semantic redesign module, in particular the visual effectiveness of the presentations adapted for the mobile device, which was ranked 2.9 ± 0.87, which is still relatively satisfactory. Examples of disturbing features were that in some cases the mobile versions allowed horizontal scrolling or the adaptation produced form elements too close to each other. However, the effectiveness of the navigation within the adapted pages was judged more positively (3.6 ± 1.07). The overall impression of the migration service was good, since the participants ranked the usefulness of concepts involved in migration highly (4.2 ± 1.03).
There were some suggestions to make the hierarchy labeling more intuitive, in order to help the user to better associate tree nodes with UI parts. Indeed, the current solution seemed more suitable for users with good background in Web technologies and with some knowledge about tree structures. In addition, two users told that it would be more intuitive to select the UI parts to migrate directly from the Web application, that is to say by clicking the HTML elements on the original page.
This section has reported on an approach for partial migration of Web applications. Examples taken from the Amazon web site have been described to show how the migration of a subset of elements is supported. The main benefits of the approach are in desktop-to-mobile migration.
The described method is based on user interface description languages and automatic reverse engineering tools able to build the corresponding logical description from existing desktop Web content. In the reverse phase, that handles all the (X)HTML and CSS tags, it is possible to identify the structure of the interface. In detail, the set of elements currently populating the page is considered, which is a fundamental issue for partial migration, because it allows to provide the user with the migratory elements.
Taking into account the limitations of the proposed solution and the users' feedback, it has been considered reasonable to design a more usable strategy characterized by direct component selection for partial migration and by a totally Web-based architecture.

4.3.2. The improved strategy for partial migration: direct components selection

The novel implemented solution allows users to directly select what parts to migrate from the original Web pages. It is supported by a Migration Client implemented as a Web application, which can thus be accessed by any type of browser-enabled device, both stationary and mobile. The Web-based Migration Client also supports migration from mobile to desktop systems and persistence across the devices of the JavaScript state during the user sessions.

4.3.2.1 Direct selection support

The interactive selection has been implemented with a mechanism included in the navigated web page. Whenever the user clicks on a certain component or a part of the page, the part "clicked" area is automatically highlighted with a different back-

ground color in order to make the user aware of the selection. It is worth pointing out that, in order to unselect a previously selected region/component in the page, the user has just to click again on the same area, and consequently the concerned part is removed from the list of components to be migrated.
The migration control panel (Migration Client) has been implemented as a Web page, thus no additional software or plugin is required on the user device in order to access the Migration Platform functionalities.
A first usability test of the improved version of both the total and the partial migration mechanism has been conducted with the support and coordination of Vodafone Italy Team, for evaluating the usability of migration, both in total case and partial one.
The study involved ten participants, which were divided into two groups: Group A performed partial migration first and afterwards total migration, Group B did the opposite. The group separation was done in order to avoid any bias due to a learning effect. In the following, a summary of the results that were gained during such tests is reported.

4.3.2.1.1. Usability Migration Evaluation - Effectiveness

The results on the effectiveness (both for total and partial migration) were quite good (see Figure 32 and Figure 33). Indeed, percentage of tasks that were successfully passed by the user was around the 99%. A few application errors occurred during partial migration: there were some problems in loading the target page. Instead, on the user's side, there was no error recorded during the execution of the task list and no user abandoned the test. Regarding the total migration (from desktop to mobile), the users always performed correctly the task list, but found some difficulties in finding certain parts of the UI, in order to continue the task on the new device. This was caused by the fact that, in case of quite large web page, the migration platform performed a semantic redesign step of the original desktop page. Thus, multiple pages were generated for the mobile device and additional links were consequently included in the pages to reach all the parts of the original desktop page. The issue was that sometimes the additional (automatically generated) links were labeled in a not very meaningful manner. In any case, no user abandoned the test.

4.3.2.1.2. Usability Migration Evaluation - Efficiency

There was no relevant difference in the execution time between the participants of Group A and Group B (see Figure 34 and Figure 35). Overall, the recorded times seemed a bit high, but this can be justified with the prototypical state of the system, that was not primarily optimized with respect to the performance.

4.3.2.1.3. Usability Migration Evaluation - Satisfaction

Satisfaction was evaluated using the System Usability Scale (SUS) technique [22] and the Product Reaction Card (PRC) method [16]. The SUS questionnaire mainly consists of a simple (ten-item) questionnaire on a Likert scale, giving a global view of the experience of the user during the testing execution. The participants provide their evaluation by answering questions regarding the system and providing scores ranging from 0 (hard to use) to 100 (easy to use). Afterwards, such SU scores are

collected and classified in terms of five usability classes (hard, fairly hard, neutral, fairly easy, easy).
The results of this evaluation, as shown in Figure 36 and Figure 37, were reasonably good, both for the partial migration and the total one. Total migration received quite good evaluations: indeed, for total migration, there was only one not positive judgment ("Hard"), which can be considered as an outlier (not caused by any platform malfunctioning). Partial migration was perceived even better.

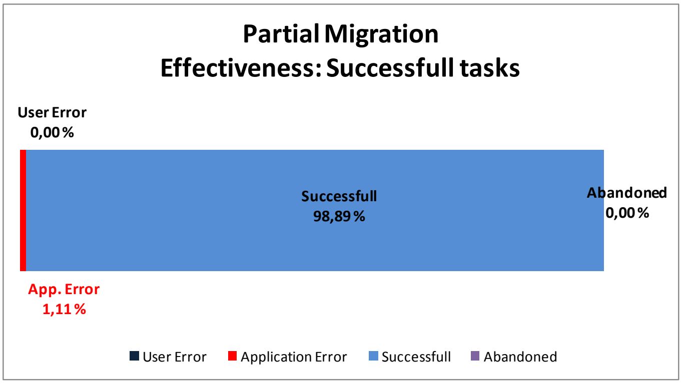

Figure 32: Results on effectiveness – Partial Migration

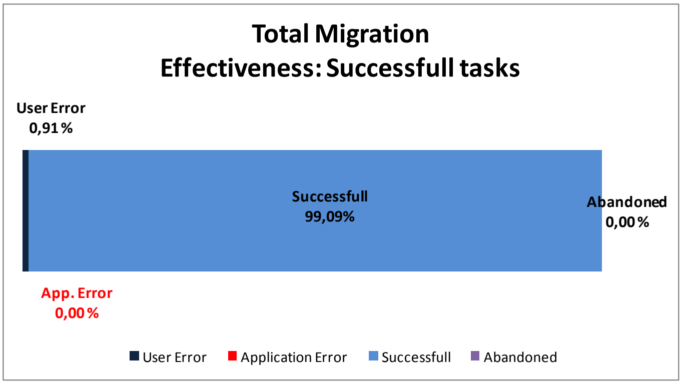

Figure 33: results on Effectiveness – Total Migration

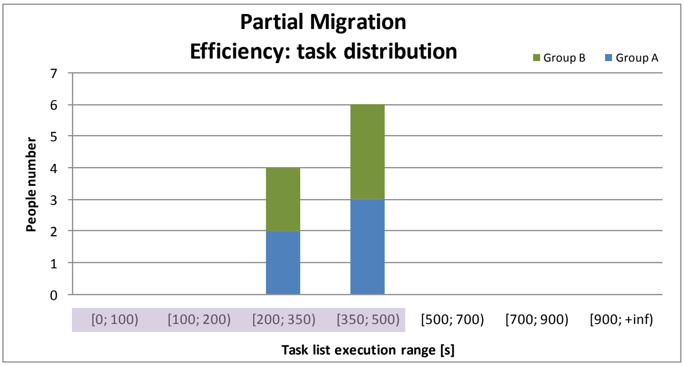

Figure 34: Results on Efficiency – Partial Migration

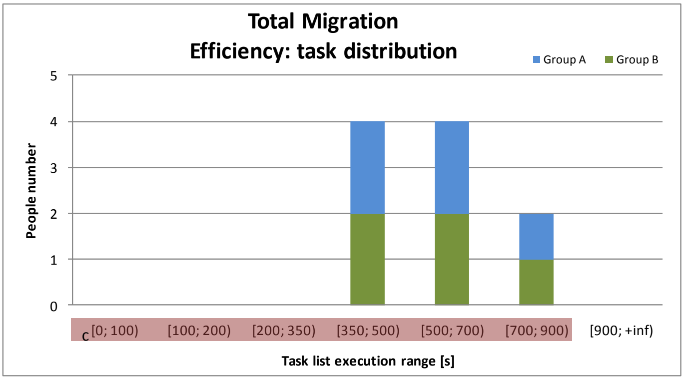

Figure 35: Results on efficiency – Total Migration

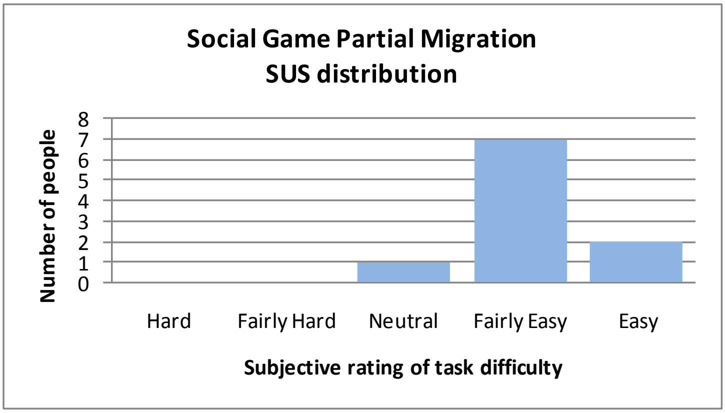

Figure 36: Results on satisfaction - Partial Migration

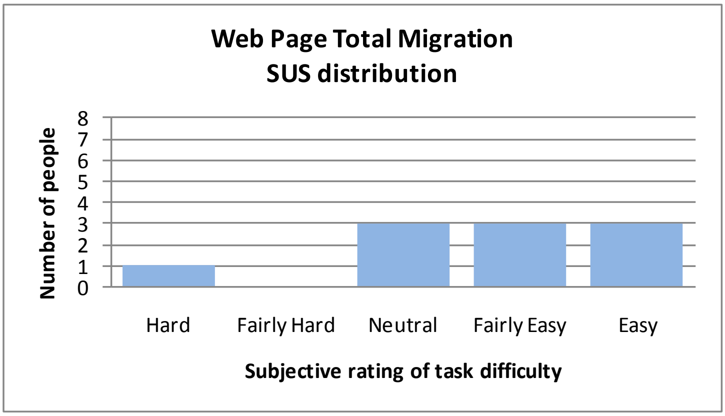

Figure 37: Results on Satisfaction - Total Migration

Product Reaction Cards (PRC) are a collection of words that can be used at the end of a usability testing, to get a sort of 'emotional response' to a prototype, etc. Regarding the PRC method, it was submitted to the users at the end of their tests both for the total and for the partial migration, as a summative evaluation of the overall migration experience. Results gained from this test are summarized on the diagram in Figure 38.

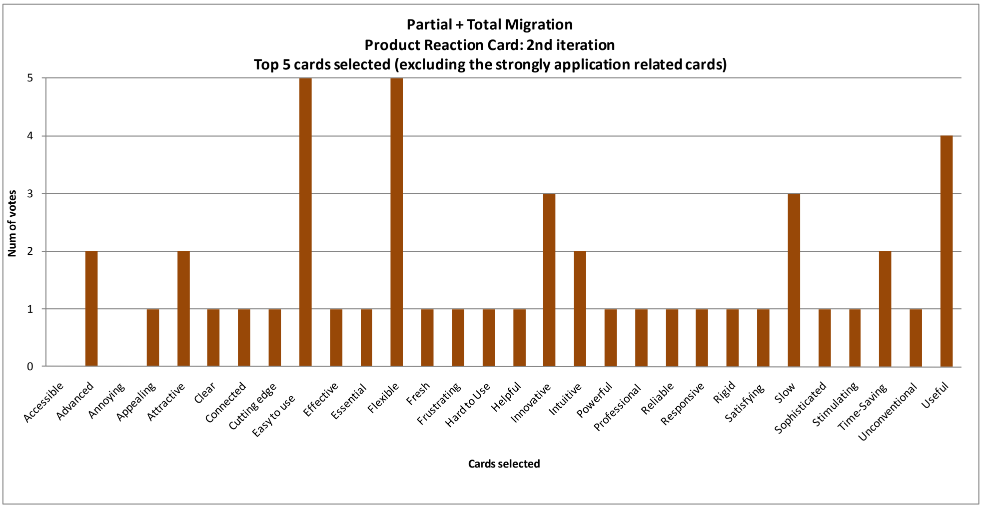

Figure 38: Results on Satisfaction of the overall Migration Experience using the Production Reaction Cards method

The top 5 cards selected by the users were: easy to use, flexible, useful, and, with the same frequency, innovative and slow. Majority of picked cards (86%) evaluates the experience in a positive manner, thus users were satisfied.

In summary, the test gained very good results with respect to all the three usability principles: effectiveness, efficiency and satisfaction.

4.3.2.1.4. Technical Migration Evaluation - Satisfaction

Apart from usability evaluation, some technical evaluations have been also carried out in order to understand the performance of the migration platform. In particular, a technical analysis was conducted so as to gain data about the performance of the overall migration platform and, down to a finer granularity, about the various modules of the migration platform (e.g., Proxy, Semantic Redesign, Generator, etc.), in order to evaluate their impact on the overall performance. The times for migration for different web pages, but also their characteristics were recorded and analyzed in order to understand to what extent the different sub-modules are affected by some characteristics.

For the various modules of the migration platform, the following parameters were considered: number and size of CSS files, number and size of JavaScript files associated with the web page, number of images, size of the considered web page, number of nested nodes in the tree corresponding to the considered page, number of links within the page, etc. Some parameters are more relevant for certain sub-modules, while other parameters are relevant for others. For instance, the number of images included in the page is particularly relevant for the Semantic Redesign: indeed, for every image included in the page, the Semantic Redesign has to check whether resizing is needed (i.e. whether image size is beyond a certain threshold). Therefore, the time taken by the Semantic Redesign is somewhat proportional to the number of images. The number of resized images also impacts the Semantic Redesign time.

Other sub-modules have different characteristics. For instance, the time required by the Proxy to elaborate a page is affected by the number of links, since it has to annotate every link included in the page.

4.3.3. Considerations and open issues

Some aspects were not managed by the Migration Platform and others highlighted the need of further improvements, especially after evaluation tests.

One point is that the Reverse engineering part is able to build corresponding logical descriptions from (X)HTML/HTML5, CSS and JavaScript implementations, but not from Flash or Java applets. When the Reverse is not able to analyze some interface part, the corresponding pieces of code are either replaced with alternative content provided by the application developers (such as images) or passed as they are to the target device.

Another aspect that was highlighted as needing improvement in the Migration Platform was the meaningfulness of the link labels that are automatically added by the Semantic Redesign module when a web page splitting is carried out. So far, the labels for such links are taken from the element name and id that are included in the page. More intelligent techniques, would aim to identify more meaningful names for such links in order to enable the user to better understand which content the linked page will actually contain.

4.4 Multi-user migration

This section describes an environment able to support users in seamless access to Web applications in multi-device contexts.

The environment supports dynamic push and pull of interactive Web applications, or parts of them, across desktop and mobile devices while preserving their state.

The mechanisms for sharing information regarding devices, users, and Web applications are described. The discussion tackles various levels of privacy and the first experiences with the proposed environment.

The availability of various types of interactive devices, their interaction resources combined with the advent of social computing, allows people to use their interactive applications together with friends and colleagues, rather than in isolation. Web applications, which are the most common, have obviously been involved by such trends, outlining the need to support shared interactions of the users in the emerging ubiquitous computing scenarios.

As the number of computing devices per person is increasing, migration capabilities are becoming more and more important. In general, there is a clear need for people to be able to exploit the technology while freely moving, especially when accessing applications requiring long sessions (such as games, e-commerce, specialized services, etc.). Pulling applications from other devices in case the user thinks it can be useful for her, is an example of how to fully exploit the technological offer.

A novel solution has been designed and developed for this purpose, able to support migration of interactive Web applications even when multiple users are sharing multi-device environments. The solution enables migration of applications (or parts of them) from one device to another with state persistence. Migration is supported through an environment that poses no constraints in terms of authoring environments to use when developing Web applications. The environment is aimed to provide flexible support in various aspects taking into account the variety of possible devices that users may want to employ.

A useful feature when accessing complex applications is the interactive selection of the parts of the user interface to migrate to the target device. This capability, as discussed in the following subsections, is particularly desirable when the destination device has limited screen size. The task requires a minimum effort, since the only requirement for the user is to run the Migration Client (still a Web application) on the browser. The Migration Client allows to log into the migration environment, and to discover what other users are on-line and the devices available for migration. The user/device visibility and availability within the migration community are managed according to privacy and device protection policies, which are set by each subscriber. Migration can occur through personal and public devices depending on such policies.

The major research question of this investigation is whether such migration platform is a usable and useful tool to better support seamless Web application access in multi-device environments. In the following subsections, after discussing related work, a couple of example scenarios in both business and personal domains are mentioned. The environment for managing migration in multi-user and multi-device environments is described, together with the associated privacy policies and including some architectural details on its components and their

relations. A user test performed with the multi-user migration environment is also described.
An initial framework for cross-platform service user experience was proposed in [106]. It was based on a study asking a number of users to access three multi-device applications for some weeks and then report their feelings in semi-structured diaries. Three main dimensions were identified as relevant: composition (the distribution of functionalities across devices is compliant with user expectations); continuity, and consistency. The study described considers access to contents and functionalities through various platforms at different times. Instead, the solution presented in this subsection aims to provide novel efficient mechanisms to seamlessly access existing Web applications across multiple devices.
Dearman and Pierce, in a study about why and how people use multiple devices [37], found out that users employ a variety of techniques for accessing information across devices. Participants reported managing information across their devices as the most challenging aspect of using multiple devices, therefore there is still room for improvement. Thus, migratory interfaces can be an relevant support from this viewpoint.
While Pick-and-Drop [93] mainly transfers data across various devices, in migratory interfaces the interactive part of an application moves from one device to another preserving the state of the user interactions. Kozuch and Satyanarayanan [62] put forward ISR (Internet Suspend Resume), a migration solution that encapsulates all volatile execution state of a virtual machine. The major limitation of this approach is the incompatibility of the virtual machine between different platforms. Indeed, the virtual machine of a desktop PC cannot be transferred, for example, to a mobile device.
A Tcl/tk toolkit for deploying distributed GUIs is presented in [73]. Differently from such work, the aim of the Migration Platform is to be suitable with any Web application developed with standard Web languages, without requiring any further, specific add-on.
Page Tailor [17] is a tool for reusable end-user customization for the mobile Web. The platform described in this dissertation integrates customization with migration, since it enables end users to decide what components to migrate to mobile devices.
Quan et al. [92] proposed to collect user parameters into an object called user interface continuation. Programs can create user interface continuations by specifying what information has to be collected from the user and then supplying a callback (i.e. a continuation) to be notified with the collected information. Differently from the mentioned approach, the migration platform supports the possibility of pausing the execution of a task, and then to resume and continue it afterwards.
Some concepts that are exploited in the Migration Platform approach are also used in Highlight [78], a server-side architecture that enables rapid prototyping and deployment of mobile Web applications. With Highlight, existing Web sites (also including dynamic ones) are re-authored to support smaller, task-specific interactions. This is achieved by embedding a Web browser inside a proxy server and using a remote control metaphor where the mobile browser controls the proxy browser. In the Migration Platform approach the server is used first to get the source of the original page and automatically annotate it with scripts for managing the migration-related functionalities.

One of the main limitations of previous solutions for migratory interfaces [15] was the fact that they were able to manage only migration for single users interacting with single applications. Little work has been dedicated to supporting multiple users in Web applications in multi-device contexts. An interesting contribution in this area was WebSplitter [52], a system for collaborative Web browsing by creating personalized, partial views of the same Web page depending on the user and the device. In that system developers had to specify the Web content in XML and define a policy file indicating what content should be shown for each device and user. Its main limitation is that it does not allow preserving the state when people and applications change device.

In the area of collaborative Web browsing, PlayByPlay [109] supports collaborations among users by means of Web page annotations. The migration platform does not address collaborative browsing, but is instead a general purpose, browser-independent environment for supporting multi-user migration, and does not require additional software installation (such as plugins, etc.).

Collaboration among users within Web applications was also previously addressed in [40]. In that system users surf the Web through a proxy server, which adds some JavaScript code to the navigated pages in order to include a toolbar to control collaboration. The JavaScript inclusion is analogous to the one performed by the migration platform. However the original layout of the page is not modified by the migration platform (which may be confusing for users): the Migration client (used for accessing the migration environment) is instead placed on a separate browser tab. Also, rather than supporting co-browsing (considered in [40]), the platform discussed here supports state-persistence, which implies that when a migration towards a new device occurs, users can continue the interaction from the point they left off while having all the data that were available on the previous device. The specific approach adopted for JavaScript state persistence of Web applications migrating across multiple devices is introduced in [15]. However, while [15] focuses on how the JavaScript state is preserved in mono-user Web migration, this subsection discusses several novel functionalities: support for multi-user migration (the application can go from one user to another); social awareness of other users/devices; pulling of Web pages from other devices; support for privacy/security.

A framework for task migration across devices, named Deep Shot, has been recently proposed [28]. With Deep Shot, an interface is migrated by simply taking a picture of it with a camera-enabled mobile phone. The support manages state persistence of the migrated interface. Some functionality of the Deep Shot framework must be manually integrated at developing time in order to make the application migratory. In addition, for Web applications a browser-specific plug-in should be installed. As already stated, the platform discussed in this subsection provides a solution accessible through any browser-enhanced device, while preserving the application state on the client side (e.g., values of form fields, JavaScript variables, cookies).

4.4.1. Example scenarios

The typical migration scenarios described in the following deal with two main migration modalities: push and pull. The former is the forwarding of Web pages from the device in use towards another device enabled to receive a migration. The latter allows users to select Web pages on a device different from the one currently used, and receive them towards the current user's device. In both cases the state of the interactive application is preserved.

4.4.1.1 Scenario 1 (professional domain)

Barbara, Christian and Marco work for a technological company, based in London, which is participating to an exhibition in Milan. Having to attend the event, they need to find a suitable flight and accommodation for the trip. They want to travel with the same flight and to stay at the same hotel. Marco starts the flight and hotel search by using the Migration Platform. Barbara and Christian, meanwhile, look at his Web activity through their Migration Client, since all of them are registered users on the platform. On the British Airways Website, Marco searches for available flights from London Gatwick to Milan. Then, he opens the Booking.com Website, specifies the dates, chooses one of the hotels in Milan that has a discount agreement with his company and selects a single room. In addition, in the reservation form, he specifies the related company's discount code and also informs the hotel that the check-in will likely be after midnight. Indeed, on the departure day all of them will have a short meeting in the afternoon and therefore Marco has just realized that with this constraint they will not be able to reach Milan before midnight. While completing the hotel reservation form, an alert window pops up in the browser, notifying Marco that Barbara is requesting to pull his flight page to her own device. Marco quickly accepts the request from Barbara. Moreover, still from the browser window of the Migration Client, Marco triggers the migration pushing of the hotel reservation form towards the devices of Christian and Barbara, and temporarily suspends his activity on these pages, waiting for his colleagues continuing the filling of the form from the point he left off.

At this point Christian and Barbara have to insert their personal details (name, email, credit card number) since some general, shared information (date, room type and additional note) had already been filled in by Marco and is preserved by the Migration Platform. In particular, Barbara, after selecting the preferred flight solution (i.e. the last flight of the day), and providing further details about tariff flexibility, insurance, and on-board meals, pushes the resulting page towards the devices of Marco and Christian. Barbara then inserts her personal details and purchases her ticket. Marco and Christian can also finalize the purchase, skipping the forms already filled by Barbara (thus saving time) while being sure to travel with the same itinerary.

In the above described scenario, users benefit from the Migration Platform features while coordinating themselves: the Migration Client indeed provides an indication of the Web activity carried out by the others. Thus, for instance, Barbara guesses that Marco's flight reservation session is ready to be pulled as soon as she notices from her Migration Client that Marco is probably interacting with the hotel reservation page (and thus has left the flight page in background).

The platform subscribers might need other communications tools (e.g., email/chat/phone) to get better coordination. However, such tools would provide support complementary to the Migration Platform which, differently from other solutions, provides state persistence and cross-device continuity for Web interactive applications.

4.4.1.2 Scenario 2 (personal domain)

Mary is at the library and surfs the technology section of an online magazine, through her laptop. She has an account on the Migration Platform, where she has registered her laptop and iPhone. She wants to browse the news through the laptop and to push some partial migration towards the iPhone. Indeed, Mary is aware that, while the laptop has a free wired connection, the mobile device is connected to the mobile network, with a tariff plan based on data traffic. Mary wants to bring only the text of the news and the menu with the social network links to the iPhone in order to minimize the data traffic. Thanks to partial migration (whose main benefit is the reduction of page size and complexity), some parts of the original page can be cut.

Mary decides to partially migrate several pages to the mobile device, so that each of them contains a short article and the social network links. Afterwards, she can leave the library and go back home (bringing her mobile device). On the metro, Mary looks at the migrated articles through the mobile device, and shares only the most interesting ones on her network. She also notices, through the Migration Client, that a friend who is preparing a master degree thesis is online and has set his device available for migration. Thus, Mary migrates an interesting article on that topic towards his device.

4.4.2. The environment

One of the principles that have driven the design and implementation of the proposed migration is interoperability, namely the possibility that several classes of devices (desktops, tablets, PDAs, etc.) can access the platform services in spite of having different operating systems, interaction resources, etc. In order to overcome this issue, the main requirement has been to support the access to the Migration Platform through a set of functionalities accessible via Web. Such environment enables users to navigate the Web through a dedicated proxy server, which automatically includes the migration capability on the visited pages. Migration triggering is also done through a Migration Client (implemented as a separate Web application). Then, the environment transparently manages continuity and state persistence of the migrated Web pages. Thus, no explicit user intervention is needed to extract the state from the source page and to restore it in the target one.

4.4.2.1 Architecture

Figure 39 provides an overview of the architecture of our environment. In order to be able to exploit the migration capabilities, users have to register their devices to the migration platform through the Migration Client.

The Migration Client is a Web application for managing the main Migration Platform functionality. It includes access to a device discovery protocol (see arrow 1 in Figure 39), and lets the user launch any Web application (2) through the Migration Proxy (3, 4), which annotates them with some JavaScript code that is then exploited to support migration. When migration is triggered by the user (5), the HTML DOM (Document Object Model) and the state of the interactive Web application are sent to the migration server (6), which updates the DOM with the whole interaction state and uploads it to the target device.

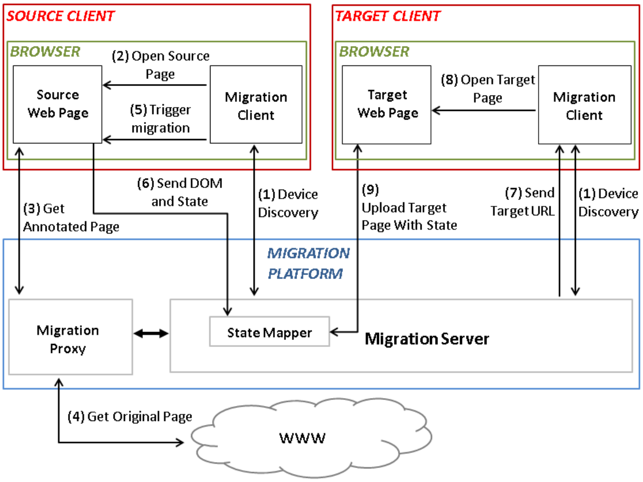

Figure 39. The Architecture of the Multi-Device Environment.

The upload is carried out in three steps (see Figure 39): first an incoming migration message is sent by the Migration Server to the Migration Client of the target device (arrow 7 in Figure 39) that specifies the target page URL within the Migration Server, then a new window/tab is activated with the target URL (8), and finally the target page is loaded from the Migration Server (9). More detail about the Proxy and the Migration Client is provided in the next subsections.

4.4.2.2 Proxy-based navigation

The migration proxy is a Web service that accepts HTTP requests towards a Web page and responds with an annotated version of that page (arrow 3 in Figure 39), by including additional JavaScript code. Such modification is needed in order to make the accessed page suitable for the Migration Platform. In detail, when accessing a Web page through the platform, the original HTML document is initially downloaded and parsed by the proxy. The additional scripts injected in the navigated pages by the proxy do not affect the original functionalities of the pages, but are activated only when migration-related functionalities are requested by the user. Some examples of additional functionalities are: the function for sending the current page DOM and the state to the migration server; the method that enables partial migration, and thus the selection of components to be subsequently migrated.

In addition, by using an Apache library (http://hc.apache.org/httpcomponents-client-ga/) the Migration Platform is able to handle secure connections with Web pages accessed through the HTTPS protocol, by saving the digital certificate in the Migration Server. More details on the security-related concerns are provided in the following.

4.4.2.3 Migration Client

As previously mentioned, some functionalities of the Migration Platform are made available through the Migration Client. The latter is a Web application that enables the users to register, login and manage their personal devices and privacy parameters, navigate the Web, and activate migration-related functionalities. By specifying the URL of a page within the Migration Client, the user opens that page (arrow 2 in Figure 39) passing through the Migration Proxy. According to user preferences, each page can be launched by the Migration Client in a new browser tab or in a separate window. This is done by exploiting the browser support for creating a new tab or window able to run in background.

The main user interface of the Migration Client (see Figure 40) provides various contextual information: the Web pages that the user is navigating, the list of the devices available in the migration environment along with indications of their current users and active Web pages (if the active privacy policy allows their visualization).

To trigger a migration, the user he has to access the Migration Client tab/window, where the list of source pages available for migration (i.e. the pages navigated via the platform proxy) is shown. Every page is identified by its original URL and title, together with the set of associated functionalities: enabling/disabling/reset of component selection for partial migration, and migration request (see Figure 40).

Upon migration request, a tiny window pops up, enabling the user to choose the actual target device(s) among the available ones (see Figure 41). It is worth noting that the only available devices are the ones that belong to the user, the ones belonging to other users which have been set by their owners as publicly visible and targetable, and the public devices.

As previously stated, the component selection on a navigated page is possible when partial migration is enabled from the Migration Client. The scripts of the Migration Client are indeed able to access every navigated page because they keep a reference to every window opened via the proxy. Thanks to such reference, the Migration Client can manage the navigated documents in order to: read the title and URL of the page, invoke the function to enable or disable the components selection, trigger the migration, etc. The Migration Client is usually accessed via HTTP protocol (HTTPS is only used when performing login to the platform or when modifying personal information). However, a particular situation occurs when a page is navigated via HTTPS protocol while the Migration Client uses HTTP.

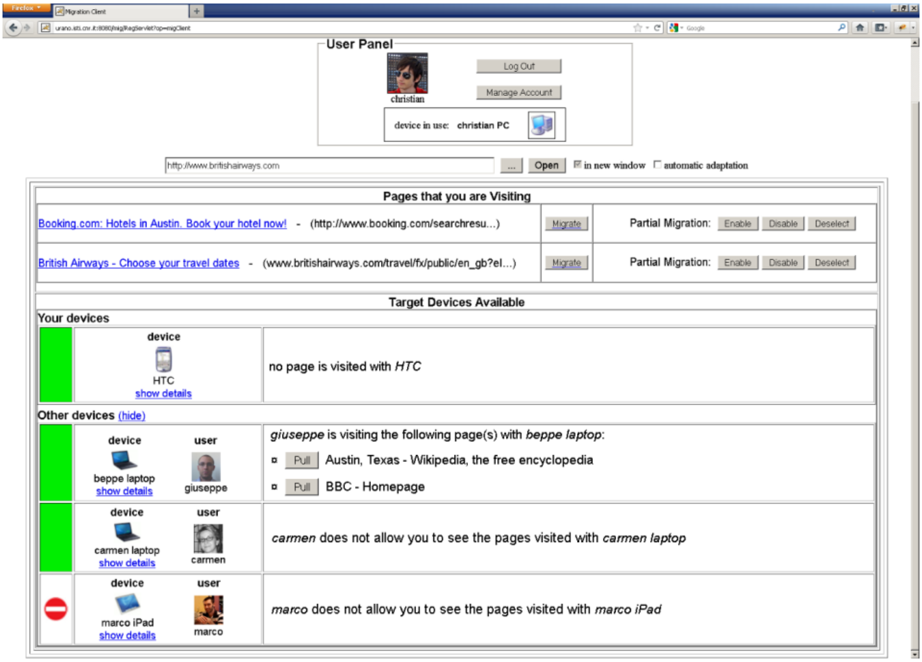

Figure 40. The Migration Client.

In this specific case it is not possible for the Migration Client to access the migratory (injected) functionalities of the navigated Web page. This is due to the difference between the two protocols, which would raise a cross-domain exception for any access. For overcoming such issue, an inter-window communication strategy has been created that exploits alternative mechanisms of the Web browsers in the cross-domain case. In this situation the Migration Client does not directly access the navigated document but uses the inter-window message-exchange support of the browser. Specific routines in the navigated pages are delegated to dispatch the messages and to perform the requested operation (e.g., enable components selection, trigger migration, etc.).

Component selection enables the user to select the main HTML elements of a page by mouse hovering and clicking: a mouse-over event on an element causes its highlighting in grey, while a mouse-click performs its selection for partial migration (which is highlighted in green). When a partial migration of the page is triggered, only the selected elements are sent to the destination device. The type of HTML elements that can be individually selected for partial migration are DIV, TABLE, FORM, IMG, and others that usually contain information.

It is worth pointing out that the scripts that manage the components selection for partial migration do not replace the original handlers defined in the original page. Indeed, the additional scripts are actually concatenated by the Migration Proxy to the ones that manage the original events of the page. This is done in order to preserve the original behavior of the page. Furthermore, after migrating parts of a page towards another device (partial migration), the user can still continue the interaction from such migrated portions.

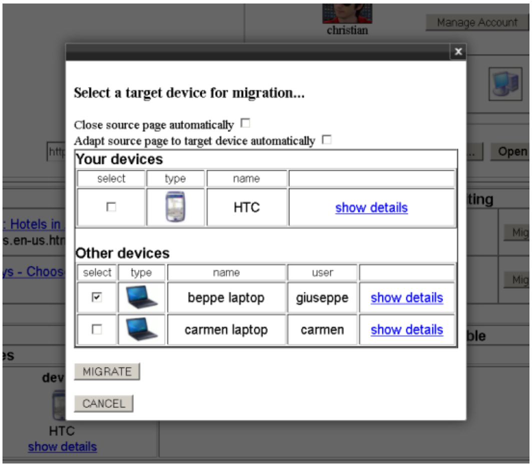

Figure 41. The Trigger Support in the Migration Client.

4.4.2.4 Privacy management

The Web interface through which every subscriber manages both her devices and the related privacy/security policies is referred to as Account Management page, which is accessible through the Migration Client. Devices can be added, deleted and configured with different levels of privacy by the Account Management page. The configuration (see Figure 42) comprises several parameters whose values indicate the level of privacy of the related device. In particular, the following parameters are defined:

- visibility: whether the device presence is visible to other subscribers of the platform;
- activity: it refers to the possibility that others can detect the list of pages visited by the device;
- migrability: it specifies whether other subscribers can address migration requests to the device;
- public use: whether the device can be publicly used, like the devices deployed in public areas.

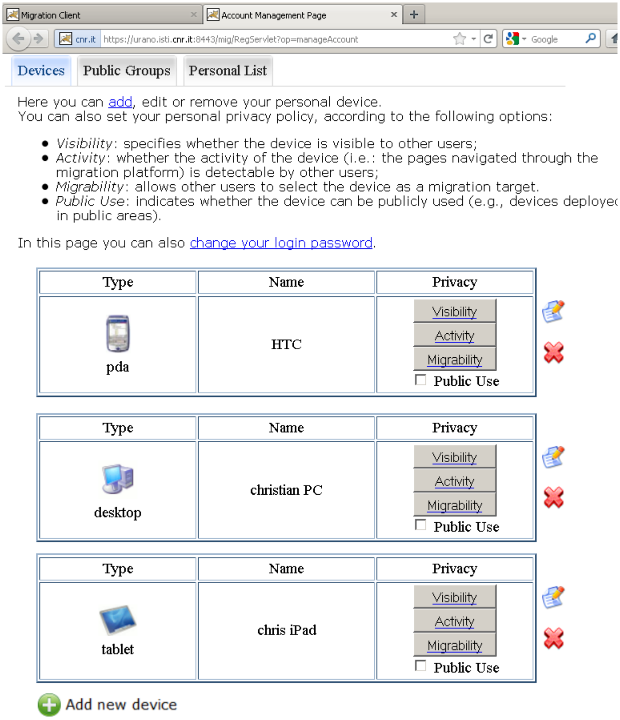

Figure 42: The Account Management page.

For each of the above mentioned parameters, several values are possible: public (everybody is allowed), private (only the device owner is allowed) or limited (only a subset of users is allowed). In particular, when the privilege is limited, one or more groups of subscribers (previously defined by the user) can access to a particular capability of the device. For instance, a user might want to configure her office desktop PC to be visible and activity-detectable from the device of every colleague; selectable as a migration target by only some colleagues, while the same device should not be even visible to other subscribers (including friends). Instead, the personal laptop could be visible, activity-detectable and migratable for everyone belonging to the friends group.

4.4.2.5 State persistence

URLs are often used to exchange pages through chats/instant messenger including some state information. However, URLs cannot provide full state persistence for most interactive pages. The Migration Platform automatically handles state persistence and provides the target device with the migrated interface in real time.

Capturing the current HTML DOM of a page just before migrating is fundamental to ensure state persistence, thus the interaction continuity in the target device. It is well known that Web pages can be dynamic: what is shown in the browser at a certain time is often different from the source document originally downloaded from

the server. Thus, the DOM resulting from some user interaction with a page can be different from the HTML DOM initially provided by the application server.
In order to solve this issue, when a migration is triggered, a JavaScript function in the page currently visited by the user is invoked by the Migration Client. This function serializes the current DOM of the page. The serialized DOM, together with the state of the forms and the JavaScript variables are then forwarded to the Migration Server, which prepares the page for the destination device. The destination page is obtained by updating the source DOM with state information. Form values are mapped into the HTML, while JavaScript variables are updated to the proper value when the target page is opened on the destination device: this is done by a JavaScript restoring procedure that gets the variable values from a file stored in the user session folder of the Migration Server. User cookies are stored in the Migration Server as well.

4.4.3. Evaluation

A user test was performed in a laboratory, with the aim of investigating both the usability and usefulness of the Migration Platform. Since the scenarios concerned a limited number of users, scalability was not among the aims of the current study, and therefore related measurement were not considered.

4.4.3.1 Test organization

Before starting the test, each participant was given a brief introduction to the system's aims and capabilities. Also, a written description of the considered scenario and the list of tasks to be performed was provided to all of them. The participants carried out the test one by one, while coordinating with the test supervisor who played the role of two additional virtual users involved in the social environment. After the trial, each user had to provide some personal information and fill in an evaluation questionnaire, asking them to rate various aspects of the proposed migration features, and to provide further comments to motivate their ratings. Each user performed the trial and compiled the questionnaire, in about 1 hour. After the test, each participant received a gadget as compensation for her contribution.

4.4.3.2 Participants

The user test involved 16 persons, aged between 25 and 51 (M: 35.3, SD: 7.3), and gender-balanced (8 females and 8 males). They were all recruited among the personnel of a research institute, but were not members of the laboratory which performed the investigation. Five of them held a Bachelor degree, 5 a Master degree, 4 a PhD and 2 a High school diploma. Although 3 users already knew other systems for sharing Web pages and links, none of them had previously used any version of the Migration Platform.

4.4.3.3 Tasks

In the test scenario, three colleagues are interacting with the Migration Platform at the same time. One of the colleagues is the test participant, while the others are the test moderator and a virtual one (emulated by the test moderator). As in the business scenario previously described, the colleagues are supposed to arrange a trip abroad and are willing to reserve the same flight and hotel.

The trial was performed by using two international, well known Web sites (Booking.com and BritishAirways.com) in their original versions (i.e. accessed on their proprietary servers via the Migration Proxy). Each participant had to interact with a laptop and with an iPhone. In detail, during the test the moderator first opens Booking.com (searching for a hotel) and then BritishAirways.com (for the flights), while the user makes her laptop able to receive a migration from everybody. In this way the moderator can push to the user a selection of convenient hotels. After accepting the migrated hotel page from the moderator, the user pulls the British Airways page from him. In the received page, the user chooses the preferred date for travel, and makes the activity of her device visible to everybody. The moderator then requests to pull from the user the current British Airways page. The user accepts the migration pulling request, continues the interaction with her own British Airways page, and then pushes this page towards the PCs of both the moderator and the virtual colleague (Christian). Then, s/he goes back to her Booking.com page and refines the search (e.g., by specifying a price range). After enabling partial migration, s/he goes back to the Booking.com page, chooses some parts of this page and triggers the migration on her iPhone. S/he then looks at the resulting page, and checks whether there is everything s/he selected before.

4.4.3.4 Results

Users were asked to rate various aspects of migration in a 1-5 scale (with 1 as the most negative score and 5 as the most positive one), also providing further comments. In the following, users' ratings with respect to various aspects of migration-related features are reported in terms of mean, standard deviation and median values. Such aspects cover various properties like e.g. usability, clarity, usefulness, and will be discussed in separate sub-sections.

4.4.3.4.1. Usefulness

Total pull migration ([3, 5], M: 4.3, SD: 0.7, Median:4)

Users thought that pulling a migration is particularly useful when working in a group, as it allows users to share resources quickly.

Single device, total push migration ([4, 5], M: 4.6, SD: 0.5, Median:5)

The total migration pushed towards a single device was considered useful for transferring a copy of the currently visited page to another user, or for migrating towards a stationary device. One person mentioned having used a similar feature on an Android device, but without state persistence since the session was not maintained.

Total push migration towards multiple device ([3, 5], M: 4.6, SD: 0.6, Median:5)

This is the case when the migration is pushed simultaneously towards multiple devices, and implies sharing information with friends or within a working group.

Participants considered the multiple migration as a quick way to notify several other people at the same time about a page containing interesting information.

Partial push migration ([2, 5], M: 4.2, SD: 0.9, Median:4), Five users stated that the partial migration is useful to select only the parts they are interested in, which is especially true when the target device is small. Two were unsure about the real usefulness, and one said that the partial migration might cause problems in old-style pages with frames.

Privacy policies ([1, 5], M: 4.5, SD: 1.1, Median: 5),
Six users highlighted the importance of privacy for protecting personal information. Four users found very useful the possibility of creating groups of users for managing the privacy levels (e.g., work colleagues, friends, relatives). Only one user stated that the privacy options are useless, since users registering to a multiuser Migration Platform should be fully aware of the fact that pages can be pushed/pulled to/from her device.

4.4.3.4.2. Usability

Total pull migration ([3, 5], M: 4.4, SD: 0.6, Median:4)
Four users would have liked small graphical improvements of the user interface for the Migration Client. One commented that the Migration Client takes some time to get used to. Another user would have preferred to have the navigated page within the same window of the Migration Client (i.e. within an iFrame).

Single device, total push migration ([3, 5], M: 4.4, SD: 0.6, Median:4)
The only relevant comment about this feature was a generic suggestion about improving the layout of the UI which could make the task of triggering such migration even more intuitive.

Total push migration towards multiple device ([4, 5], M: 4.6, SD: 0.5, Median:5)
This aspect received quite good ratings and did not receive further comments.

Partial push migration ([2, 5], M: 3.9, SD: 1.1, Median: 3.5)
The main criticism of partial push migration concerned some difficulties in selecting a component for the first time. An exemplary case is the unwanted click on a link while the user actually meant to just select a related UI part for partial migration (e.g., clicking on a menu item when trying to select the whole menu), which caused opening the linked page. One user was concerned with the selection being too dependent on the Web page structure, also declaring that, in some cases, it was not possible to exactly select the desired part, but only a larger part. This observation is actually true. Indeed, one of the main characteristics of the proposed support is that it does not affect the original Web page structure. Thus, the components selection is done according to the original structure of the page: the more the page is structured, the finer the selection can be.

Privacy policies ([2, 5], M: 3.8, SD: 1.0, Median: 4)
Some issues were raised about the complexity of the user interface for modifying the privacy parameters visualized in Figure 42 (users felt that too many parameters were included together in a single page) and the lack of visibility of the selected values (the currently selected options are invisible until the user enters editing mode). It was also recommended to integrate the settings panel with meaningful icons.

Six users declared that the parameter names (e.g. visibility, activity, migrability) were not sufficiently intuitive.

4.4.3.4.3. Other aspects

Clarity of migration pulling notification ([2, 5], M: 3.9, SD: 0.9, Median:4)

As for the notification of the incoming pulled page, two users found it confusing since they had explicitly requested that page from another device.

Notifying the user that a new window/tab is being opened should be a default feature. However, it could be made optional in further versions of the platform.

Four users were concerned about the position of the incoming pulled page confirmation within the interface of the Migration Client, which was judged not immediately visible. Recommendations included centering the confirmation box and using a blinking style for highlighting it.

Preferred default migration modality

Two participants did not provide any preference for the default migration modality. Three users preferred partial migration with multiple targets. However, eight users (the relative majority) believed that the total push migration towards a single target is the most useful modality. The detailed preferences of the users in this regard are reported in Table 3.

user	Preferred Migration Modality						Notes
	Source Selection		Triggering Mode		Target		
	Total	***Partial***	***Push***	***Pull***	***Single***	***Multiple***	
1	●		●		●		
2	●		●		●		
3	●		●		●		
4				●	●		
5	●		●		●		
6							"It depends"
7		●				●	
8	●		●		●		
9	●		●		●		
10							No answer
11						●	"It depends"
12	●		●		●		
13							
14	●				●		Pushing / pulling
15		●				●	Pushing / pulling
16		●	●			●	

Table 3. The preferred default migration modality of the users.

Further migration options to consider

Users were also asked about possible migration options to consider in future development of the platform. The possibility to save a page, send it as email attachment and to see a preview of a page before pulling it were cited.

Preferred default privacy policies

Among the 16 participants, only one did not specify any default preference for privacy policies. The others provided at least one preferred configuration. To summarize the diverse combinations of parameter values declared by the users, it seems reasonable to group them into three categories based on the overall privacy level:

- Low: the presence and the activity of the device are visible to everyone; the device is targetable for migration and it can be publicly used (like the devices deployed in public areas).
- Medium: the device presence visibility is public or limited to some users (e.g., a group); the other parameters are public, private or limited.
- High: the device presence is not detectable by any other user (apart from the owner), thus no operation can be performed on it by others.

According to the provided answers, it can be stated that 7 users would choose a medium privacy level, 5 a high level and 3 a low level.
Table 4 reports the detailed preferences declared by the users. With regard to the "Level" column, "L" stands for low privacy level, "M" for medium and "H" for high.

user	Preferred default option				
	Visibility	**Activity**	**Migrability**	**Public Use**	**Level**
1	Private / limited	Private / limited	Private / limited	False / limited	M
2	Public	Private or Limited	Limited	False	M
3	Private	(Private)	(Private)	(False)	H
4	Private	(Private)	(Private)	(False)	H
5	Public	Public	Public	True	L
6	Not specified	Not specified	Not specified	Not specified	-
7	Private	(Private)	(Private)	(False)	H
8	Public	Public	Public	True	L
9	Public	Private	Limited	False	M
10	Private	(Private)	(Private)	(False)	H
11	Limited	Private	Private	False	M
12	Private	(Private)	(Private)	(False)	H
13	Limited	Private	Limited	False	M
14	Public	Limited	Limited	True	M
15	Public	Public	Public	True	L
16	Limited	Limited	Public (limited!)	True	M

Table 4. The privacy levels preferred by the users.

Applications for which migration is more useful
In general, the migration support was considered useful for a number of applications both in private life and in business. One user suggested that he would exploit it during the planning of a trip, or when developing templates for a Web site in order to share every choice with the other stakeholders involved. One user observed that a scenario in which migration can be useful is a business one similar to the one of the test, and dealing with a secretary having to prepare a travel itinerary and to find accommodation for several colleagues who are leaving for a business trip. In this case, the exploitation of the Migration Platform would lead to a significant time saving, because every colleague should only insert the personal information (such as name, phone, credit card number) before proceeding to the final reservation/purchase.

Weak/Strong points of the overall support
Among the strong points there were the innovativeness of the platform, the possibility of working asynchronously with other users and then share the results,

the state preservation after migration, the speed and reliability of the support. Privacy level configuration was also appreciated.
The weaknesses were mostly related to the user interface layout, which elicited a number of different comments. One user stated that there were too many options in the privacy settings, another one suggested putting both the Migration Client status and the settings management in a single page, while a third one would have preferred a more appealing layout.
The development of a browser specific add-on was recommended only by one user. In general, users appreciated the possibility to access the platform by any standard browser, without having to install any application or plugin.

4.5 Security and privacy in migration

As previously mentioned in this chapter, migratory interactive applications allow users to perform their tasks continuously across various devices. This flexibility of access implies various security issues, and this section discusses them and how they have been addressed in a client/server-based Migration Platform for interactive Web applications.
In current multi-device environments it is important to support flexible access mechanisms, which should consider that users often need to move and would like to opportunistically exploit the devices that dynamically become available. Application migration is a type of multi-device support in which users can dynamically change device and still continue to perform their tasks from the point they left off. As discussed in the sections of this chapter, environments supporting migratory interactive Web applications allow users to dynamically push and pull them from one device to another for various reasons. Migratory user interfaces, while moving across devices, preserve the state of their interactive part as well, which can be modified directly by the user without updating the server-side of the application. In addition, such environments can even allow the direct selection of the parts of the interactive application that users would like to migrate to the target device. However, such migrations can raise various security risks, concerning the theft of private information in the interactive applications or the intrusion of bogus (malicious) versions of the interactive migrating application to replace the original ones. For example, a user can enter personal information and sensitive data while booking a room on a mobile device and then can migrate to a desktop application (see Figure 43) with the risk that the data are stolen in some way.
This section analyses the possible security risks in a client/server architecture for migratory interactive Web applications. The design and implementation of a number of techniques that allow users to avoid security risks in such Migration Platform are discussed. A validation of the proposed solution with some well-known existing Web applications is also reported.

Figure 43. Example of migration involving personal data.

Han and others [52] presented a solution to support multi-user access to Web applications through various devices, but the associated security issues were not addressed at all. Making the actual security support in Web applications access more user perceivable was addressed in [72], where the authors propose a Firefox plug-in to include a bar whose color indicates the actual security level. The issues of security when Web applications are accessed in multi-device environments were not addressed.

Some interesting issues about access to public devices are provided by [98], which considers the usage of public fixed displays and mobile devices together. While private information can be hidden from public displays, it can be displayed on the mobile ones (which also support user input). The Migration Platform aspects discussed in this section does not tackle the issues of showing information in public devices. The discussion is rather focused on the risks associated with dynamic migration of interactive Web applications.

An architecture that affords mobile user greater trust and security when browsing the internet (e.g., when making personal/financial transactions) from public terminals at Internet Cafes or other unfamiliar locations is presented in [97]. This is achieved by enabling Web applications to split their client-side pages across a pair of browsers: one untrusted browser running on a public PC and one trusted browser running on the user's personal mobile device, composed into a single logical interface through a local connection, wired or wireless. Information entered via the personal device's keypad cannot be read by the PC, thwarting PC-based key-loggers. Similarly, information displayed on the personal device's screen is also hidden from the PC, preserving the confidentiality and integrity of security-critical data even in the presence of screen grabbing attacks and compromised PC browsers. The drawback of such a use of two devices for accessing a Web application can generate usability issues. Hence, the different solutions for preserving both usability and security have been investigated within the Migration Platform, with respect to the access to Web applications in pervasive environments.

4.5.1. Migration in a Client/Server architecture

This work focuses on a client/server based architecture for supporting migration of interactive Web applications. Figure 44 provides a high-level description of how the migration architecture works. How previously stated in this chapter, the Migration Platform allows any browser-enabled fixed or mobile device to access any interactive Web applications through a proxy, which annotates them with some JavaScript code that embeds migratory capabilities. Any device involved in the migration environment must execute the Migration Client, which is a separate Web application able to provide information on the available devices. The information is gathered and circulated through the execution of an Ajax-based discovery protocol. The Migration Client of each active device periodically queries the Migration Platform server for the available information (e.g., list of possible target devices, devices activity). At the same time, every device updates information about its status to the Migration Platform. The Migration Client is also used to trigger the migration and to select the target device for migration.

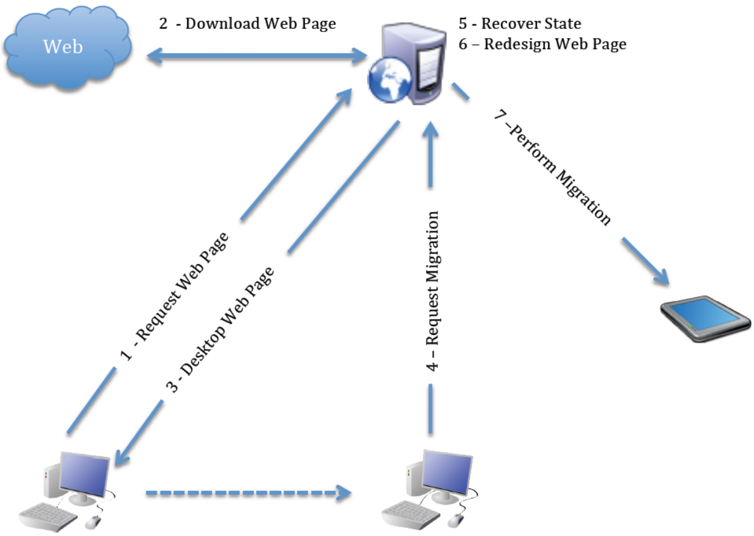

Figure 44. High-level description of the migration architecture.

The Migration Platform under discussion is based on a server, which provides a set of services, implemented as Java servlets. The main server components involved and their interactions are shown in Figure 45.

Each subscriber connects to the Migration Platform Server through his/her client device. After performing the authentication on the Login page, the user is allowed to navigate the Web via the Proxy. The Web pages are enhanced with migratory capabilities by injecting scripts that, upon request, send the Document Object Model (DOM) of the page to the Migration Servlet. The migration process is terminated when an instance of the source DOM is launched on the target device.

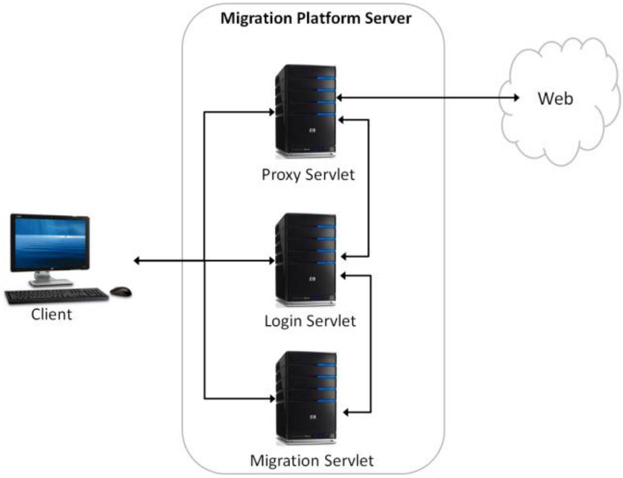

Figure 45. High-level description of the migration server components.

In general, migration can be:

- *single-user*, when a mobile user decides to change device and continue to interact with the same application in the target device;
- *multiuser*, when an interactive application is moved to a device accessed by another user (either through push of the user on the source device or through pull of the user on the target device).

The multiuser migration of a Web page, as discussed in the following sections, introduces several potential security issues which are tackled in the following.

4.5.2. Security issues in the Migration Platform

As a starting point to identify the security lacks of the Migration Platform, the most known types of attacks that threat user security and privacy can be considered. In the following discussion both the lacks that were found and the proposed solutions are tackled.

4.5.2.1 Authentication attacks

The goal of an authentication attack is to skip the system authentication procedure in order to perform unauthorized operations. Such kind of attack can be carried out by several strategies, that are listed in the following.

Eavesdropping strategies consist of listening to the packets exchanged between client and server, in order to get the user access credentials. Eavesdropping is among the most serious threats of the systems that do not exploit login encryption. The initial Migration Platform prototype suffered from the eavesdropping risk, as it was initially deployed on a Web server which supported only HTTP connections. It was thus possible for anyone, especially in public networks, to steal the user credentials by simply sniffing the login packets exchanged between the Migration Client and the Login Servlet. To overcome this issue, the Secure Socket Layer (SSL) has been exploited, as explained later on.

The access credentials may be stolen even by repeatedly querying the system. A similar strategy, known as *Brute force*, can be performed by iteratively trying the login with different credentials, until the right combination is found. A protection against the brute force attack is limiting the number of access attempts that can be performed within a specified amount of time.
Another way to trick the authentication mechanism is the *code interpretation* of scripts within a Web page. When the decision of what information to display according to user privileges, or even the authentication mechanism is delegated to the page scripts, an authentication attack can be performed by simply interpreting the scripts. Scripts interpretation, in this case, reveals the system communication logic. Any modifications can then be made to the page/scripts in order to skip the controls, and to access the system without any authentication. This issue affected the first version of the Migration Platform, where the authentication control was delegated to the JavaScript of the Migration Client Web page. The JavaScript, in that case, managed the login control deciding whether or not to show the server responses (i.e. the list of other users and available devices). Therefore, since the server always responded to the requester, an attacker could have created a modified version of the Migration Client to monitor the activity of the subscribers without having the privileges. The solution for the Migration Platform has been to move the whole authentication mechanism to the server, which properly responds only to requests from authenticated clients.

4.5.2.2 Session management attacks

Such strategies, often referred as *session hijacking*, aim to steal the user session identifier (that is often sent as a cookie) in order to act as the user. According to [107], session hijacking is among the most common threats today, and can be carried out by the following mechanisms:

- *Cross site scripting* (XSS) is the theft of cookies performed by malicious scripts injected in the navigated page.
- *Session sidejacking*, similarly to eavesdropping, consists of listening to the network packets with the aim of extracting the session cookie(s).
- *Replay attack* is performed by sending to the application server copies of packets previously “stolen” from the network. The packets, which contain the original cookies, are used to commit frauds (e.g., repeated transactions).

The Migration Platform was initially exposed to cross site scripting, session sidejacking and replay attack. However, by means of several security improvements, only the external side communications (i.e. the communications between proxy and application server) could be affected by such kind of attacks, as explained in the following.
Session fixation is done by forcing a user to use a known identifier to query the system. This type of attack typically exploits the forwarding of the identifier as a URL parameter, thus it is not a concrete threat for the Migration Platform (which does not allow a session id to be passed in this way).

4.5.2.3 Authorization attacks

Systems typically perform controls on user privileges before allowing the access to functionalities and resources. The authorization attacks aim to trick such controls.

In the Migration Platform, such threat affected the private user data (e.g., name and type of personal devices, list of navigated Web pages) as well as the device functionality. Regarding the latter issue, an intruder could have been able to interfere with the user device functionalities by overriding the controls on incoming migration privileges specified by a user. Thus, the intruder could migrate a malicious Web page towards the user device. This attack, as discussed later on, is unlikely to be a threat for the improved version of the Migration Platform, and user operation requests are now formally validated.

4.5.2.4 Information theft and privacy violation

Personal information, such as credentials or navigation chronology, can be easily stolen if data are exchanged transparently between client and server. As previously discussed, this issue is unlikely to affect the Migration Platform since the private data exchange between Migration Client and Login servlet are encrypted.

4.5.2.5 Code injection

A malicious code fragment can be injected into a system by passing it as a parameter through some exposed method. If the input parameter is not properly validated, code injection can be extremely dangerous. A widely known code injection technique consists in passing a query string to the form for querying a database (e.g., a search form). The code injection applied to the system database could even return the entire database, including personal data of the system subscribers.

In the Migration Platform, the subscribers' data are stored on an XML database, which is not subjected to code injection. However, code injection can also affect the proxy servlet for navigation. This could be done by passing an inconsistent URL as a parameter of the GET method and exploiting some weakness in the control. For instance, if a bad string is passed as URL to the proxy and the exception stack is printed (e.g., in the error page), the attacker can even obtain information about the system source code. A set of additional controls on the input URL consistence have been integrated in the proxy in order to improve its robustness. The exception stack trace printing has also been disabled.

4.5.3. Migration Platform operations security

The main Migration Platform operations, such as login, profile update, migration request, are explicitly performed by the user through the Migration Client. Other operations, such as the device discovery messages (e.g., user/device presence) are periodically and automatically sent by the Migration Client to the platform server.

The aim of the login is to protect user personal data, and information and resources of other subscribers who are in relationship with the user (and to which intruders would not be entitled to access otherwise).

The login on the platform is carried out via HTTPS, thus protecting the user credentials against theft. However, as discussed in the previous subsections, intruders may also exploit session sidejacking by listening to the network packets and extracting the session cookies from the headers. A possible way to overcome this problem could have been to perform all the operations via HTTPS. However,

differently from the login operation (which is performed only once per session), the device discovery protocol operations are repeatedly performed and must be carried out in real-time. As a consequence, though the solution is technically feasible, it would have heavy drawbacks on efficiency.
Regarding the operations allowed to registered users, a non-secure and unique session cookie, referred to as JREG, has been exploited. The uniqueness of JREG is aimed at improving the security. Since the cookie can be used just once, an intruder is very unlikely to capture the cookie and reuse it to act as the owner user. The uniqueness is achieved by appending an increasing timestamp to the cookie. This is done by the platform that, upon user login, calculates the JREG by concatenating the deviceID and the output of the HMAC function (1):

deviceID **|** HMAC (deviceId, timestamp, session id, k) (1)

where:
- *deviceID* is the unique identifier of the device (created at registration time);
- *timestamp* refers to the instant of the last received request;
- *session id* is the identifier of the actual session;
- *k* is the encoding key for HMAC generation;
- *HMAC* stands for Hash-based Message Authentication Code.

The usage of a hash encoded function is necessary to assure the following features:
- confidentiality: since data sent are private, it is better not to send them unencrypted;
- integrity: if the cookie is modified by third parties, the hash function applied to it returns a different value. The deviceID modification is an example of possible attack attempt;
- uniqueness: one and only one encoded string refers to a specific character sequence.

The strategy for improving cookie security reported on [67] is similar to the one above mentioned.
It is worth highlighting that it is possible for an intruder to use the stolen cookie before the actual owner. As a consequence, the system considers the access of the real user off-time and authenticates the intruder. The proposed Migration Platform, in order to limit such risks, sets the HttpOnly attribute to the JREG cookie, thereby making it inaccessible to malicious scripts. Although the risk of replay attacks still exists, the cookie theft by means of XSS attacks is avoided.
An additional strategy, aimed to limit the consequences of cookie theft, exploits a complementary cookie named JSECURE. At login time, the servlet generates the JSECURE cookie, with secure attribute set as “true”. The secure attribute lets the cookie be exchanged only via HTTPS. Operations involving the exchange of personal user data, such as the profile update, are always performed via HTTPS. For each of those sensitive operations, the system checks the consistency of the incoming JSECURE cookie. Due to the check, the intruder is unable to perform any malicious operation as s/he does not know the actual JSECURE value. The real

user would instead be able to re-login at any time, gaining full access to the system.
The session cookies are generated as message digests so as to achieve uniqueness and unpredictability. The SHA-256 algorithm of the “javax.crypto” package is applied to a random number generated through SHA1PRNG algorithm of the Java “SecureRandom” class of the “java.security” package.
It is worth pointing out that, as already stated, although the above mentioned strategies do not completely exclude the possibility of attacks, they tend to limit the consequences.

4.5.4. Proxy and navigation security

When dealing with the security of the proxy-based navigation within the Migration Platform, several issues arose that needed to be addressed. For the sake of clarity, such problems can be studied by dividing them according to the communications side they refer to: the external side and the internal side (see Figure 46). Those referred to as external side issues are related to the communication between the proxy and the servers that host the original applications (i.e. the ones which the clients request access to). The internal side comprises the issues that spring from the data exchange between client and proxy.
In the following subsections, the two categories of problems and the proposed solutions are discussed in detail.

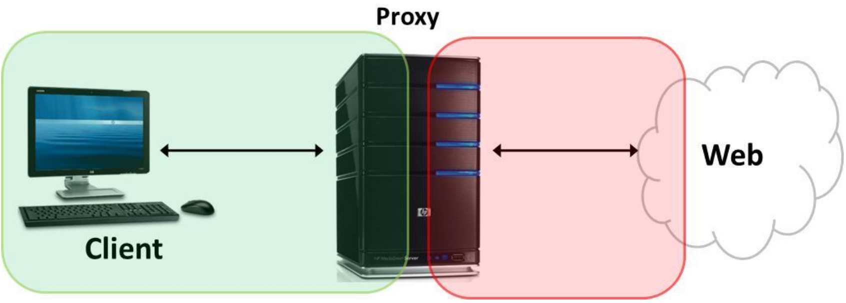

Figure 46. The two sides for the navigation security.

4.5.4.1 External side

The proxy-to-server communication issues were tackled at first, as it was initially clear that the proxy had to be able to perform secure connections in compliance with the application server. This is because Web applications often exchange data with clients via the HTTPS protocol, though their domains are usually accessed via HTTP. The protocol change from HTTP to HTTPS is done by the application server in order to protect sensitive user data against third party sniffing. The user login credentials as well as credit card details, for instance, are typically sent via HTTPS.
The Migration Platform proxy is able to connect to any server via HTTPS, as well as via HTTP. The connection is managed by the *org.apache.http.client* library that fully implements all HTTP methods and supports encryption with HTTPS protocol

[58]. In practice, the level of security in the external communication is determined by that requested by the original application.

4.5.4.2 Internal side

On the internal side, security involves the communications between the client device and the proxy. An exhaustive description about the internal side security is given in the following subsections.

4.5.4.2.1. Navigation security

An extreme strategy to provide navigation security would consist in always connecting to the proxy via HTTPS. This solution might technically provide the highest security, but is considered to be inefficient since the usage of HTTPS is known to decrease data rate. A study on the performance impact of the SSL is reported in [110]. Thus, the Migration Platform proxy has been enhanced in order to “switch” to HTTPS depending on the application server. The proxy is then accessed via HTTPS only when the application server requires it.
The proxy has been improved with HTTPS capability by installing a X.509-compliant certificate in the host web server. Also, an additional port (i.e. the 8443) complementary to the conventional HTTP port has been set up.

4.5.4.2.2. Cookies management and security

Several strategies have been considered and tested for the management of user cookies within the Migration Platform.
A first strategy consisted of making the proxy forward every Web application cookie to the client. The name field of the cookie was annotated by appending the original domain (i.e. the domain of the application server). The client browser could then locally store every cookie, while the proxy did not keep any of them. The main drawback was on the client: indeed, for the browser, all the updated cookies were actually referred to the domain of the proxy. Thus, the client browser appended all the proxy cookies to every subsequent request to the proxy, which restored the original cookie names and only forwarded the cookies belonging to the domain of the application server. It is worth noting that this strategy is highly inefficient.
The improved mechanism for cookies management that has been developed is totally proxy-based. The application cookies are never sent to the client but are instead saved in a cookie store entity, which lies in the proxy. The cookie store keeps the name/value/domain of each cookie and the reference to the owner client (which is considered to be a subscriber to the Migration Platform). For every client request, the only cookies forwarded by the proxy to the application server are the ones associated to the requested domain. Each client is bound to its previously stored cookies by a proxy session cookie, which is a token generated by the Web server hosting the proxy and which is stored on the client browser. The header of every client request to the proxy contains the proxy session cookie. The proxy uses it as an index to the cookie store section of the user.
In the following, application server cookies are referred to as external cookies, while the proxy cookies to as internal cookies.
The theft of an internal cookie would allow an intruder to access all the external cookies of the user. As a consequence, the intruder could exploit the cookies to recover the user session on the application server of banks, email, etc. A possible

option to protect the internal cookie would be to always exchange it via HTTPS. However, as already mentioned, this strategy heavily impacts the navigation efficiency.
The usage of a temporary unique session id is also unsuitable for the proxy, because several pages can be navigated at the same time on the client (i.e. the user may open more than one window/tab for navigating via proxy). The possibility to use the client IP address as unique user identifier has also been excluded, since the device IP can change dynamically in some cases. The Migration Platform is indeed accessible by mobile devices, which often switch from one network to another (thus changing their address).
The navigation cookie, which identifies the user session, has been chosen to be non-secure and constant, but additional controls are performed by the platform. For every request, the proxy checks whether the user-agent of the browser is consistent with the one stored at login time and, if not, the request is rejected. After this first check, in the event a resource is requested via HTTPS, the proxy performs an additional control on the JSECURE cookie which was set up at login time. Since the JSECURE cookie is secure, it is highly protected against thefts, thus the intruder is unlikely to know it.
HTTPS navigation via proxy occurs, as previously stated, only when the application server requires the SSL usage (i.e. when private data are going to be exchanged). Thus, it is reasonable to assume that sensitive data is safely exchanged via the Migration proxy.

4.5.4.2.3. Security of shared resources

The possibility of sharing information and Web interactive applications among several devices, is among the main features of the Migration Platform. The sharing of Web interactive applications is also referred to as "functionality sharing" because a Web page typically hosts functionalities, that often may be accessible only to the authenticated user (e.g., email Web client, bank account management, …).
When a Web page is migrated, a copy of the page is created and stored on the Migration Platform server. When the migrated page is ready, the Migration Client of the target device is notified in order to open it through the browser. A technique for notifying the target device is to send back, via Ajax, the reference to the migrated page, which is actually the URL within the Migration Platform server. This solution was adopted by the first Migration Platform prototype, since it was considered to be simple and effective. In detail, the URL string was contained in the response of the Ajax request periodically performed by the Migration Client, which updates the device presence and checks for incoming migration pending. An initial security analysis revealed the risk of passing a URL through an HTTP (thus unencrypted) response. Indeed, an attacker could get the URL by simply listening to the network communications, and then use it to access the migrated page on the Migration Platform server.
As previously discussed, the proxy navigation is considered to be secure. Indeed, when HTTPS connections are performed via proxy, the secure cookie created during user login is used. Thus, even if the intruder got access to the migrated page, s/he would not be able to perform secure navigation, because s/he does not own the right secure cookie. However, the consequences of a similar attack lie in the possibility of the intruder to perform unauthorized operations or acquire

sensitive data even from the single migrated page. It should be assumed that the migrated page could contain reserved information, such as the balance of a personal bank account. It is thus desirable for the Migration Platform to protect the Web pages during and after (as well as before) the migration process.
To protect the migrated pages from unauthorized access, a dedicated web service has been developed. The service provides the (migrated) target page content as response. In detail, instead of the migrated page URL, a special command is passed back to the Migration Client. Such a command triggers the invocation of a special service, named "loader", which loads the target HTML document and writes its content over the response stream. The Migration Client of the target device gets the response content and shows the related page on a new window. Such a procedure is carried out only after successful authentication of the target client, as the loader service will actually provide the page content only if the JREG cookie of the client is consistent with the one related to the actual target. The target device is indeed registered and active (i.e. the user has previously logged in).
A requirement of the migration process is the state persistence: when the migrated page is opened in the target device, the result of the interactions performed in the source device has to be maintained. The page state includes form fields content, JavaScript variables value and the cookies related to the Web page domain. As already discussed, although the external cookies are not stored in the client devices, the user can indirectly access them through the Migration Platform. The proxy is indeed aware of the correspondence between the user and the original cookie(s) value for the requested domain. When a migration is triggered, the external cookies of the source internal session are bound to the target internal session. This is implemented by creating a reference, in the cookie store, from the internal session of the target device to the internal session of the source device.
From the point of view of the users involved in the migration process, two situations can occur: the single user and the multiuser. Major security drawbacks could arise from totally copying the cookies in the case of multiuser migration, i.e. when the application migrates to a device owned by another user. In this case the target device can access all the cookies owned by the user of the source device. In detail, the user accessing the target device not only accesses the cookies related to the migrated Web page, but every cookie that was bound to the user accessing the source device since the beginning of the session.
There is more than one possible alternative solution to protect source cookies in multiuser migration, according to the level of protection, as discussed in the following.

- *Excluding any cookie from migration*. The source user might prefer to migrate the Web page, but not the associated cookies. Only the content of the page would then be accessible to the target user. For instance, a shopping cart page could be shared with friends or relatives through multiuser migration, but no additional operations (e.g., cart modifications or purchase) would be allowed to the target users. This solution has been discarded as it implies a reduction of the possible functionalities.
- *Copying only the cookies of the migrated page*. When the only functionalities to provide to the target are the ones within the migrated page (thus excluding the

ones of other pages of the domain), the only copied cookies are related to the page path. A user might want to provide a colleague with the copy of an interactive Web page that exploits a specific path-related cookie to preserve its interaction state (e.g., the sorting settings for the items). At the same time, cookies related to other pages of the website domain, would not be migrated in order to protect the privacy of the user. This solution is pretty unsuitable in real cases, since most web applications do not specify the path for cookies.

- *Copying all the cookies of the domain which the migrated page belongs to*. This less restrictive option provides the target with all the cookies within the domain of the migrated page. An example scenario involves the subscriber of some service (user A) that needs to temporarily allow a friend/colleague (user B) to access it. User A performs authentication on the website and migrates the resulting page to the device of user B. It is then possible, for user B, to navigate throughout the website, since s/he has got all the cookies for that domain. User B would not be entitled to modify user A credentials, as s/he would be requested to insert the original ones (which s/he does not know). This is the solution adopted by the Migration Platform.

4.5.5. Migration of personal data

Personal information is often contained, in different ways, on a Web page that is being migrated. Such data can be in the document source code, such as in the summary page of a shopping cart, or can have been explicitly entered by the user during the interaction with the page.

As previously mentioned, migration is carried out by firstly serializing the Document Object Model (DOM) within source device browser. The DOM actually includes the values of the form fields filled in by the user. In order to understand how this affected the security of the preliminary Migration Platform, it is sufficient to consider the following simple scenario.

The user is creating a personal account on a social network, through non-secure connection, while s/he decides to trigger migration towards another personal device. The migrated page contains the partially filled form, including the chosen username and password. Thus, although the form action value is "secure" (i.e. an HTTPS address), sensitive data would have been exchanged without any encryption between the client device and the Migration Platform. This is because the form page was accessed through HTTP, even if the form submission would have forced the browser to switch to a HTTPS connection.

In the improved version, at migration time, the Migration Platform is able to automatically detect the presence of sensitive data and to act accordingly.

- If the page is navigated through HTTPS, then migration is performed through HTTPS.
- If the page is navigated through HTTP, a search for secure forms is done within the document. If the page has one or more secure forms and if at least one input field is filled, then a secure migration is performed.

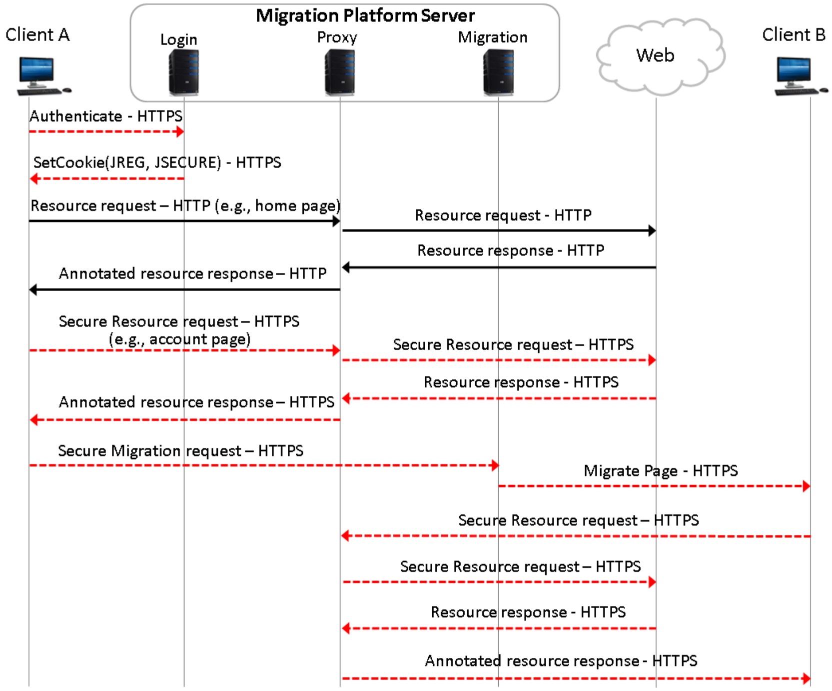

Figure 47. A sequence diagram showing the main communications involved in secure migration.

The security control on the migrated page is performed when migration is triggered. The control is carried out by the scripts injected in by the proxy at navigation time. The motivation of distinguishing between secure and non-secure migration lies in the performance: secure migration is indeed more time-consuming, and thus it is reasonable to migrate via HTTPS only in the case there could be security concerns.

Figure 47 summarizes the main communications that take place when a Web application is navigated via proxy through HTTPS, and when it is migrated. Black continuous arrows indicate HTTP communications, while red dotted arrows refer to HTTPS ones. User A initially authenticates in the Migration Platform and requests the Website home page. While performing login in the home page, s/he is redirected to a HTTPS address and continues navigating the personal pages via proxy. Upon request, the platform performs a secure migration, thus acquiring the source DOM from the source device and forwarding it to the target device through HTTPS. The user then continues the navigation through the target device, still via proxy within HTTPS connection.

A technological issue, known as cross-domain exception arose when a secure migration was triggered for a non-secure page. The reason was in the different domains between the page and the secure address of the migration servlet (HTTPS protocol, 8443 port): the page is indeed accessed via the conventional platform address (HTTP protocol, 8080 port). In order to solve this issue, the CORS (Cross-Origin Resource Sharing) support [34] has been integrated in the

Migration Platform. CORS defines a W3C standard specification for authorizing a browser window to exchange messages (e.g., through Ajax) with different domain pages.

4.5.6. Validation of secure migration

Some of the most visited web pages (according to the ranking of http://www.alexa.com) were considered and tested on the Migration Platform, with the aim of validating its secure support. The following list summarizes the operations performed on the tested Web applications:

- *google.com (Gmail)*: the login page and the email composition on the user personal page were migrated, preserving access credential, cookies and email form content (email title and body).
- *eBay.com*: the account creation page was migrated, and the user data (such as name, surname, username, password) were maintained.
- *amazon.com*: the shopping cart content page was migrated with cookies persistence (selected items were maintained).
- *wordpress.com*, youtube.com: the login page was migrated preserving user access credentials.
- *paypal.com*: the form for sending a payment was migrated, and the cookie as well as the form field content were maintained (payment amount and beneficiary).
- *facebook.com*: the login page was migrated with the user credentials form content.
- *ryanair.com*: the flight reservation form, including user personal details (name, surname and address) and credit card data, was migrated.

After each web migration, navigation was performed in the target device, in order to verify the full state persistence (in particular with respect to the cookies). Although many of the tested pages were in the cross-domain situation (i.e. pages in non-secure address but migration performed via HTTPS) the migration worked in all cases.

4.6 Migration Platform and logical dimensions for multi-device user interfaces

This section deals with a framework for describing the various design dimensions that can affect applications accessed through multiple devices, including when various users are sharing them. Such dimensions are discussed with respect to the Migration Platform previously proposed. In particular, it is described how the platform is able to support them, and some example scenarios of use are provided about Web applications.

A study reported in [37] aimed to achieve a better understanding of why and how people use multiple devices in their everyday life. The authors found out that users employ a variety of techniques for accessing and managing information across the various devices, but at the same time they reported that there is still room for improvements, especially as far as the user experience is concerned. Indeed, participants in the study reported managing information across their devices as the most challenging aspect of using multiple devices. More generally, since the number of devices per person is steadily increasing it is important to reach a better understating of how to change the way to interact with applications exploiting multi-device user interfaces.

For instance, the recombinant computing approach has been proposed and investigated with the aim to facilitate users in exploiting multiple technologies in a composite manner (rather than in isolation). An example of recombinant computing infrastructure is Speakeasy [77], which defines a framework that supports usage of resources in a networked environment. The approach does not require the involved components to have mutual awareness, but has several constraints. For instance, the components have to specify how they exchange information. This implies that the involved components must have a recombinant implementation.

In recent work [106], the authors investigate the key elements that characterize the User eXperience (UX) when users exploit Web-based applications through different computing platforms (mainly desktop and mobile devices). This study also identifies an initial framework for cross-platform service UX, in which the central elements include i)fit for cross-contextual activities (the structure of the application across different devices matches the user's activity, leading to an effective fit for tasks in different contexts), ii)flow of interactions and content (the transitions across the devices are experienced as fluid and connected), and iii)perceived service coherence (the application and its components are perceived as consistent and coherent, as part of the same service).

The framework proposed in this subsection is more structured, and is also able to highlight the main technical issues in multi-device user interfaces.

Demeure et al. [38] have investigated distributed user interfaces (DUIs) according to their definition and classification. The introduced reference framework defines four possible dimensions for the distribution: what is distributed (computation), who is distributing it (coordination), when (communication), from/to where (coordination). According to the authors, the framework does not consider the collaboration among users since this dimension is assumed to be "a natural extension" of DUI when several users are involved in the distribution. Although the Migration Platform prototype has not distribution capabilities, some analogies can be found with that framework [38]: the computation dimension, which introduces the notion of splittability, defines which parts of the interface can be distributed and is similar to the "granularity"; coordination is related to the "trigger and activation type"; communication corresponds to "development time". Regarding coordination, the framework of this dissertation introduces a refinement, since it deals with it according to "UI duplication" and "Device Sharing". However, the main difference with the framework in this dissertation (and the associated platform) addresses multi-device user interfaces in more general terms (e.g. as resulting from migration, copy, etc.) and is not limited to DUIs.

CAMELEON-RT [4] is another framework concerning multi-device environments. The problem space is defined by a few dimensions with the purpose of proposing an architecture reference framework, while this dissertation provides a more detailed set of dimensions in which the architectural approach is itself another dimension, with different possible values.

Myngle [100] is a support that facilitates device change (e.g., desktop to mobile) in Web navigation. Myngle provides an easy way to revisit previously seen content without having to cope with the specific URL of the desired resource. The support is proposed as browser extension and as native application for a few mobile platforms, thus it lacks portability. Also, it cannot support access to resources that imply the existence of further state information (such as session cookies, JavaScript variables, …) that is not usually mapped in the URL.

Multi-device applications are important in many domains. In [Jimenez 2011] the authors have investigated how to improve learnability in ubiquitous systems such as multi-device museum guides. These authors claim that one design principle to promote the learnability of such interactive systems is to improve their UI consistency. In particular, the authors identified three types of consistency, whose definition was suitably adapted in order to better fit the multi-device/multi-user context: within-device consistency, which occurs when the UI design is consistent with the design of previous applications developed for a specific mobile device; across-device consistency: it occurs when the UI design is consistent with the design developed on other mobile devices; within-context consistency: when the consistency principle is applied to ubiquitous systems, one aspect that should be taken into account is the context including aspects that are not strictly connected with the devices. For instance, a museum is identified as a social, informal context in which interactive learning is supported, therefore the user interface should reflect these characteristics to some extent. The authors have compared three different interaction styles (gamepad controller emulation, mobile multi-touch, and Wii-based emulation) and they concluded that the use of a within-device consistency generally gives better results when considering museum applications. The authors pointed out that these results can be also generalized to other similar ubiquitous contexts.

Other related work that analyses the issue of users managing multi-device applications was carried out by Denis and Karsenty [39], who introduced the notion of inter-usability to identify the ease with which users can reuse their knowledge for a certain functionality when switching to other devices.

Previous work by the research community has coped with multi-device access to applications. Olsen et al. [82] studied techniques to combine multiple clients working on the same task, and have introduced the concepts of “Join” and “Capture”. Join refers to collaborations with other users, whose clients can subscribe to data associated to a particular task. Clients are promptly notified and, whenever data changes. Capture consists of assembly of interactive resources to address specific problems, for instance by exploiting different modalities, with the aim to improve interaction. The authors assert that their integrated approach of multi-client interaction and multi-user collaboration is able to provide synchronization among tasks. This aspect can be related in some way to the persistence or “continuity” dimension of the framework which will be discussed later on in this dissertation. However, the mentioned work particularly focuses on the user mobility aspect of nomadic interaction. The multi-device access and the collaboration among users are seen as a consequence of the user mobility. On the one hand, with respect to the possibility for a user to join a task, the algorithm of the server-based strategy for task synchronization is reported to be robust and reliable. On the other hand, the paper does not consider a scenario where multiple users, independently from their physical position, share application interfaces (or parts of them) in pushing mode (i.e. by making them migrate towards peer devices). Thus, collaboration among users seems to be an implicit consequence of the explicit join of multiple users to the same task, which can be the typical situation of co-located individuals such as in a meeting room. The authors also state that, although the multiple device involved share the same network and are integrated in the same task, they are not aware of each other. Differently, in the proposed Migration Platform, mutual user/device awareness is a means to stimulate and support collaboration among users (that might not necessarily be co-located).

In general, a variety of research contributions is found in the area of multi-device user interfaces, and it is not possible to mention all of them. However, despite the increasing interest, when designers and developers want to support users in accessing applications (or parts of them) through various devices they often experience difficulties in identifying the possibilities and aspects that should be considered. Therefore, this subsection first suggests a logical framework in order to describe the range of possibilities that multi-device user interfaces can offer, by identifying several dimensions that have been judged relevant for such systems. Then, the architecture and implementation of the Migration Platform is tackled showing how it relates to the proposed framework. Afterwards, some examples and usage scenarios of the Migration Platform are proposed in order to better show its potentialities. Finally, some summary remarks and indications for future work are provided.

4.6.1. A logical framework for multi-device user interfaces

A number of factors have been identified that enable the analysis, design and comparison of multi-device user interfaces. Indeed, there are many ways to support cross-device applications. They can range from accessing them through one different device at different times (one device at each time), to situations in which the users access multiple devices at the same time through a user interface that dynamically changes its distribution across them. This framework aims to systematically analyze the various possible situations in which cross-platform access can be performed.

Cross-device applications could even involve multiple users. Therefore, in those scenarios, it is possible to have multiple devices either used in a coordinated way by the various users, or used one by one to access the application at different times. Other cases could occur when the user changes the interaction device and the user interface supports (or not) the possibility to preserve the interaction state.

These dimensions are described in the following subsections, in which various types of distributed and migratory User Interfaces (UIs) are considered.

Distributed UIs are characterized by UI elements distributed across more than one device. While UI migration occurs when the UI is first rendered on one platform and then there is a change of the device on which the UI is currently rendered, while preserving the interaction state in the target device. It can be useful to point out that distributed UIs and migratory UIs are two partially overlapping categories. Indeed, there may exist distributed user interfaces which are also able to migrate, but there are also only distributed user interfaces (which do not migrate at any

time) or even migratory user interfaces that are not distributed across multiple devices.
It is worth pointing out that the values on one of the various dimensions are not always completely independent of the values appearing on the other dimensions, since sometimes some particular values in one dimension could imply specific values in other(s). For instance, as discussed later on, if there is UI distribution between multiple devices, this implies that the granularity of the UI manipulated across the different devices cannot be at the level of the whole UI, but should be at the level of groups of UI elements, or even at a finer one.

4.6.1.1 UI distribution

This aspect analyses whether the solution considered is able to support the distribution of the user interface elements across various devices at a given time. The UI distribution implies the existence of some coordination across the involved interactive devices supporting the access to the application logic by exploiting input/output from/to the various devices involved in such a UI distribution. An example is when people interact with large screens using gesture to select some elements and voice to indicate how to manipulate them. An approach for dynamically distribution of user interfaces at run-time is discussed in [19]. The authors highlight the availability of diverse devices that characterize networked smart environments. Among the potentialities of distributed user interfaces in such contexts, there is the possibility for the user to enhance the interaction, for instance by increasing the communication bandwidth.
Although not mentioned explicitly as distribution, an early example of interface splitting among several devices was also tackled in [75]. Pebbles SlideShow Commander allowed to control a PowerPoint presentation running on a laptop through a handheld with wireless connection.

4.6.1.2 Continuity

This dimension analyses whether users are enabled to change the current device in use (the source device) and then have available the application on a different device (the target device) while the system automatically preserves the interaction state reached with the first device and offers an adapted UI on the new device. As said before, this happens during UI migration. According to [106], in particular with respect to Web service access, continuity is considered to depend on how well the system supports cross-platform transitions, task migration and synchronization.
In general, the state of the interactive application that can be captured and preserved depends on the implementation environment considered. In Web applications, for example, it can include the state of forms, JavaScript variables, cookies, sessions, history, bookmarks, etc. Other types of applications, such as those supported by cloud computing, are able to preserve the state of only server-side information. This is an optional aspect in multi-device systems, thus some of them do not provide any support at all to UI continuity.

An automatic solution for migrating user interfaces and preserving their state has recently been presented [28]. This approach, called Deep Shot, allows the migration of an interface (or part of it) by simply “shooting” it with a mobile phone camera. The authors claim that Deep Shot is compatible even with applications that are not Web based. However, some extra work by the developers is needed to enable deep shooting/posting within an existing application. The platform described in this subsection currently supports Web applications, but does not require any modification to existing Web pages or services because it exploits a proxy server that automatically adds the necessary support on the navigated pages.

4.6.1.3 Granularity

This parameter considers the granularity of the user interface that can be manipulated across various devices, for example through distribution or migration. At least four levels can be distinguished:

- *entire user interface*: the user interface can be seen as a single monolithic item, which can be moved/copied from one device to another;
- *groups of user interface elements*: structured parts of user interfaces (e.g. navigation bars, articulated content areas with text and images, …) are distributable across various devices;
- *single user interface elements*: single UI elements are distributed, for some reason, to other devices;
- *components of interface elements*: interactive elements, which are usually characterized by a prompt, an input, and a feedback are distributed across various devices. For instance, the user enters an input through a mobile device while the resulting feedback is shown on a large screen.

Some levels of granularity were addressed in previous sections of this chapter, where partial/total migration capabilities of the platform are described. Various levels of granularity were also addressed in [73] for supporting user interface distribution.

4.6.1.4 UI duplication

In multi-device user interfaces, a number of possible choices can be identified to manage the ability to duplicate the user interface (or even parts of it) displayed in the current device, for example to better share the current content with other users. This is the case when there are two devices in which two instances of the same application are active at the same time. This could be helpful when the user might want to activate multiple, subsequent (namely: separated in time) migrations from the same source device. For instance, a user might want to migrate from device1 to device2 and then, after a certain time, s/he might want to migrate from device1 to device3: therefore, the Web application should not be closed in the source device in order to enable the second change. There are also some intermediate cases. For instance, when the user selects just a portion of the UI to be moved onto another device s/he can decide to keep active on the source device the whole UI or just a portion of it. Therefore, there could be a total UI duplication, a partial duplication or no duplication at all between the involved devices.
Multibrowsing [60], which allows users to move existing pages among multiple displays, can be considered as an example supporting user interface duplication.

4.6.1.5 Trigger activation type

This dimension analyses how the request for a change in the cross-device user interface is triggered. This change could then activate e.g. a migration or a (re)distribution of the UI. The simplest case is user-initiated, in which the user actively selects when, to which device and what should be changed. With automatic trigger the system autonomously activates the change when it recognizes that some conditions are verified. This can be done, for example, in case of a high battery consumption level and a simultaneous user's proximity to another device that has power supply. Therefore, the system might decide that a device change is appropriate and then select the new device to be used. This type of automatic trigger can be related to the work on implicit human-computer interaction driven by the context discussed in [95], in which the system acts proactively on the basis of context information. There might also be a mixed approach in which the system first automatically suggests a change to the user, while still enabling her to modify some parameters in the request. Therefore, in this case there can be a mixed type of trigger activation (partially suggested in an automatic way and partially edited by the user).

4.6.1.6 Device sharing among multiple users

This aspect considers when multiple users are involved in the multi-device user interface environment, in particular whether there are devices that can be shared by multiple users. This can happen either because the same device is targeted by the user interfaces of their applications (an example could be when two users use the same large display as a target for a migration from their mobile devices) or different users access the same interface on the same device (e.g. when two or more users exploit the same wall-sized interactive screens). The multi-user/device framework discussed in [40] and the related scenarios explicitly consider the situation of several users accessing the same interface on a public display concurrently.

Sharing implies that the supporting environment should be able to indicate what the shareable devices are, whether there is any conflict in their use, and provide some information regarding their state. Needless to say, in some solutions this possibility could not be supported.

4.6.1.7 Temporal effect

The time when a device change should occur in the multi-device configuration is considered by this dimension. The most typical case is a migration directed towards a target device, which has to be carried out as soon as the migration trigger is sent from the source device (immediate effect) in order to achieve seamless continuity. The other case covers the possibility for the user to specify the time when to defer the change in the multi-device configuration towards some device (deferred effect). This could be useful when the target device is temporarily unavailable to the user, hence the effect will be delayed until a better time. It is worth pointing out that in this case the support should provide users with appropriate mechanisms to choose the device to be used as target, even if this device is temporarily unavailable in the current environment. The deferring time is implicitly managed in Deep Shot [28], that allows to launch on a different device an application with a previously captured work whenever the user needs it. Myngle [100] could also be seen as an example of system that manages in some way the temporal effect, since it lets the user choose when to restore the previous state of a Web application on a different device.

4.6.1.8 Interaction modalities involved

This dimension analyses the modalities involved in the multi-device user interface. Mono-modality means that the devices involved in the cross-device access support the same interaction modality. Trans-modality means that, after the device change, the user has to change the interaction modality. Multi-modality occurs when the multi-device interface simultaneously supports two or more interaction modalities in one (at least) of the devices involved. Various interaction modalities have been considered in the dynamic interactors distribution proposed in [19]. The authors assume that each device category available on a smart environment has specific interaction resources (IRs). The approach is based on distribution of interface interactors among available IRs (thus devices). Distribution is performed automatically according to context information and developer/user settings. In their work, the authors refer to multi-modality as the combined use of multiple modalities within a (distributed) user interface. However, they do not explicitly report on supporting more than one modality within the same device.

4.6.1.9 User interface development time

This dimension specifies when the user interface is obtained so as to be rendered on the target device(s). On the one hand, in the design-time case the user interface has been built in advance for each type of device. Then, at run-time only the state has to be updated, in case of migration. On the other hand, there is the case of run-time generation of user interfaces, where a run-time engine dynamically generates -at the time the device change occurs- the user interface, according to the target device features. An intermediate approach is also still possible where the supporting engine dynamically generates the user interfaces for the different devices by exploiting some logical descriptions which have been created at design time.

The aspects related to the design time are also discussed in [105], which presents Dygimes, a testbed for model-based user interface development. Networked cooperating devices potentially offer a set of interaction resources to the mobile user. The authors distinguish between static and dynamic approach for UI distribution, i.e. for distributing interaction resources (IRs) among UI components. The static approach would require to know at development time the runtime context peculiarities, which is rather difficult since it implies to known which IRs are available in the environment and to which IR every part of a UI will be distributed. The dynamic approach consists of allocating UI components to IRs at runtime, either automatically by the system or manually upon user request. The authors are aware of the issues due to classical UI design: while dynamic splitting is difficult, creating different implementations of the same UI part for every possible distinct IR and interaction device is very inefficient. Thus, they propose as possible solution to describe the UI on a model-based manner. Different models are used to create descriptions at development time, while the actual user interfaces are generated at runtime starting from the higher-level models.

4.6.1.10 Adaptation aspects

When the device currently in use is changed, some adaptation is often required in order to support the user interface. Various levels can be impacted by the

adaptation process: either the entire application can be changed depending on the new context, or just some UI logical parts (such as presentation, navigation, content) or even single UI components can be adapted. The case that no adaptation is provided also exists. However, if the type of device is changed, the generated results often provide low usability.
Adaptation at the presentation level can refer, for example, to possible changes in the presentation layout. There are various ways to adapt the presentation, ranging from simple scaling to applying information visualization techniques (e.g. semantic zooming, fisheye, …).
Navigation adaptation involves the connections among the different presentations: for example, when the number of presentations increases or decreases then the connections between them are adapted accordingly.
Content adaptation happens when some information is removed, added, or modified (e.g. summarized) in order to produce a more usable user interface depending on the resources of the device. The already introduced Platform for partial migration of Web user interfaces has adaptation capabilities, which are particularly useful when Web pages are migrated towards small devices: pictures are scaled, content is split in several presentations, interaction components are replaced by more suitable ones.
The multi-user/device framework previously cited and discussed in [40] also tackles interface adaptation: transformation modules are used to convert information into different representations (e.g. visual to audio) according to user's preferences. Transformations rely on previously defined annotations that specify how to convert resources at runtime.

4.6.1.11 Architecture

With regard to the architecture of the supporting platform two different strategies can be considered: (i) client/server, in which there is an intelligent unit managing all requests and sending all data to target devices, thus controlling the user interface allocated across various devices; (ii) peer-to-peer, where the devices directly communicate and negotiate the distribution parameters. Within this type of coarse-grained distinction, it is still possible to identify more refined approaches for managing some phases, one of this being the interaction state preservation (when supported). For example, with a client-side approach there can be some client-side mechanisms, such as plug-ins or applets, that gather the interaction state data on the client side and then send the entire dataset to the migration engine for further processing. With a server-side approach there is instead a server which gathers each event occurring on the client side and consequently stores suitable information. It is worth considering that the choice between client/server and peer-to-peer may be driven by other aspects of the whole platform. For instance, if interface adaptation features are included in the platform, then the server support is highly desirable, since the computational effort needed to transform/generate the target interface might be too huge for a mobile device. The Migration Platform previously described in this section is based on a server which creates logical descriptions of the source interface, adapts it to the target device capabilities and generates a specific implementation on-the-fly.

The cross-device infrastructure of DeepShot [28] also relies on an instant messaging protocol which is server-based, even if the devices involved in migration are usually co-located.
On the contrary, a peer-to-peer strategy could offer more flexibility. For example, a set of devices equipped with peer-to-peer clients in the same network that do not rely on an external Web server could exchange information locally. This simplified architecture also leads to lower communication latencies between devices. An example of peer-to peer implementation for distributed user interfaces is in [73], which requires the use of a specific development toolkit in order to benefit from this feature.

4.6.2. Migration Platform and multi-device framework support

4.6.2.1 Technologies and motivations

This subsection describes how the previously described Migration Platform supports a considerable part of the space defined by the dimensions of the above introduced framework. The Migration Platform extends various aspects of previous work. The implementation choices that led to the development of the Migration Platform for multi-device Web user interfaces were driven by interoperability, flexibility and usability motivations. The Platform is server-based. The core support is hosted by the services that provide proxy navigation, state persistence and Web application multi-device capabilities. The strategies and implementation details of the module that provides state persistence for pages containing JavaScript are discussed in [15]. Such services are accessible from a gateway referred to as Multi-device Control Panel. A similar strategy to coordinate Web pages moving among multiple devices was proposed in [60], and was referred as multibrowsing. In that case, however, the control panel was implemented as a browser plugin for a specific browser (thus for a single platform). The control panel of the Migration Platform is instead implemented using standard Web technologies that make it suitable for any Web browser. The main advantage of such configuration is that a Web browser is the only requirement for the user device to successfully access the Platform capabilities. Most of the currently available devices (both stationary and mobile) are Web browser enabled, and thus they are compatible with the Platform.
The Platform has been successfully tested through PCs/Laptops/Smartphones with several operating systems and browsers, e.g. Windows Mobile PDAs, iPhone/iPad. It is worth considering that the access to the platform capabilities does not require any prior installation of software (such as applications or plugins) on the user device, and this aspect favors usability.
The platform is able to manage any Web application written in a standard language, without knowing its code in advance and without requiring any prior modification. Thus, web developers are not required to explicitly enhance the Web application with migratory capabilities and can use any authoring environment. The pages navigated by the users are instead automatically parsed and enriched with migratory functionalities by the platform proxy server.

4.6.2.2 Platform architecture and dimensions supported

An overview of the platform architecture including the main communications between the modules, is depicted in Figure 48.

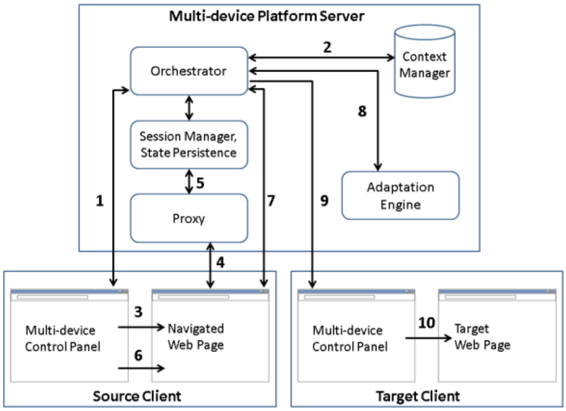

Figure 48. The Architecture of the Platform.

In order to illustrate how the architecture works, it is worth considering a typical use case of a Web application in a multi-device environment. First, the page of the (Multi-device) Control Panel (which is an enhanced version of the previously described Migration Client) is launched on the user device (1). Such Control Panel is a Web application, hosted by the Multi-device platform server, which allows the user to log in the platform and exploit its services. After verifying user/device credentials (2), the Control Panel lets the user open a new Web application via the Multi-device Platform Proxy (4) and a new session is started (5). The user interface of the accessed Web application, which has been enriched with multi-device capabilities by the Proxy, is shown on a new browser window/tab. The reference to the new window/tab is held by the Control Panel, which is able to trigger and control the multi-device interaction, when it is needed. Examples of multi-device interaction are migration of the current application towards another device, distribution of the application or a part of it among several devices, etc. Such functionalities are controlled through the Control Panel, which communicates with the Platform and with the window/tab of the navigated page without affecting its original layout.
Interaction between the Multi-device Control Panel and the navigated page is carried out through inter-window JavaScript function calls (6). If adaptation of the source page is needed (e.g., when the page is being migrated or copied from a desktop to a small device), the Adaptation Engine (8) is called by the Orchestrator (invoked by the navigated page) (7) in order to create a version of the page suitable to the destination device.

The reference to the resulting page, i.e. the URL of the target page, is then sent to the Control Panel of the destination device (9), which opens the reference and shows it in a new window/tab (10).
The adaptation process is supported by the use of model-based languages [45], which provide logical descriptions of interactive applications at various abstraction levels, thus facilitating interoperability across various implementation languages. In

this Platform, the logical descriptions are dynamically generated through a reverse engineering module when the page is accessed.

Information about active devices is managed by the a module that manages context of use information (Context Manager), and may include: type of the available devices, owner, activity and security/privacy level. The device activity is represented as the list of Web pages currently displayed on the device. The privacy level is reflected by how the context information is shown (or hidden). For example, if the user does not want to allow other users to know which applications are active in her device, then s/he can keep the application list as “private”. The highest level of privacy is achieved when the device is invisible to other users, while an intermediate level consists in having the device visible but private and not available as a target device for other users.

Managing security and privacy within the platform, as discussed in the previous section, is not only a matter of how information is stored and circulated, but is a more general problem that has driven the implementation choices of the whole platform. An example is the proxy support for Hypertext Transfer Protocol over Secure Socket Layer (HTTPS). The HTTPS is automatically provided by the platform proxy whenever it is required. Indeed, it has been assumed that, when performing an online task with security concerns through the multi-device platform, a user should be provided with at least the same security level of the original application server (e.g., mail server, support for paying with credit cards, etc.).

Table ***5*** summarizes whether and how the proposed platform supports each particular dimension of the framework for multi-device user interfaces mentioned in the previous subsection.

Framework dimension	Platform support
UI Distribution	Not supported.
Continuity	Supported by the state persistence of the web application (HTML forms, session cookies, JavaScript variables).
Granularity	Addressed by the total or partial migration option: the whole web page or some components can be migrated, according to user needs.
UI Duplication	Supported as long as the web page on the source device keeps running after migrating towards the target device.
Trigger and activation type	Manual migration triggering is available so far. Automatic triggering based on the context of use (e.g., location change) is enabled though a web service allows for migration triggering.
Device Sharing between multiple users	Device sharing may occur during multi-user migration, i.e. when a user interface (or a part of it) is migrated towards the device of another user. Consequently, within a target device, an arbitrary number of interfaces belonging to different users can be deployed.
Temporal effect	Migration is supported seamlessly, in real time. It is not possible to defer migration to a given time.
Interaction modalities involved	Multimodal migration is not supported.
User Interface Development Time	The migrated user interface is completely generated at run-time.
Adaptation aspects	Supported by the adaptation engine, which exploits models to abstract the logical descriptions of the source web page and creates an implementation suitable for the target device.
Architecture	Client/Server: mainly based on the multi-device platform server, but with some support (e.g., the Multi-device Control Panel) running on the user device. Most interaction state data is gathered on client-side and forwarded, at migration time, to the server, but some information lie in the platform proxy server (e.g., session cookies).

Table 5. Summary of how the proposed Platform supports the multi-device framework.

4.6.2.3 Example scenarios

Two examples of how the Platform can support users in multi-device access to Web applications are provided in the following.

<u>4.6.2.3.1. Migration from stationary to mobile device</u>

Barbara is leaving her home for a holiday trip to Brussels. Before going to the airport, she visits a Web photo gallery through the desktop PC and finds an interesting photo album about the main sights and monuments of Brussels. She intends to bring to her mobile device only those pictures that could be useful to

decide which places and monuments are worth visiting. Hence, Barbara enables the Partial Migration option from the Migration Client on her PC and selects some parts of the page: the thumbnails of the favorite pictures and a piece of the upper navigation menu. Then, she makes them migrate towards her mobile device. Through the partially migrated page, during the taxi transfer and while waiting for the flight at the airport, she will zoom in on the pictures, read the descriptions, and plan her Brussels tour.
The desktop page of the Web photo gallery, together with the selected elements (left), and the page migrated through the Platform (right) are shown in Figure 49. The selected elements are highlighted by green frames on the source page and their composition is rearranged by the adaptation engine before being sent to the target device.

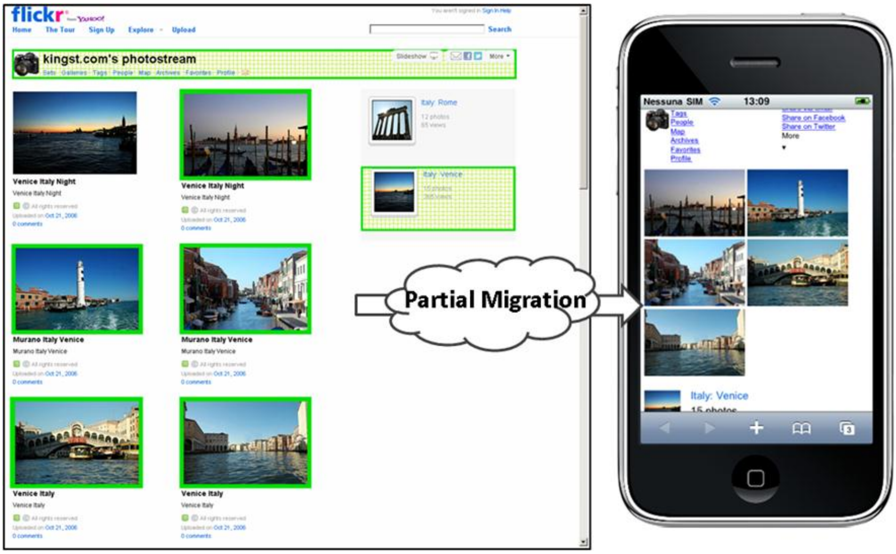

Figure 49. Example of User Interface Migration Supported.

The Platform provides support for the partial user interface migration through some scripts that are injected in the navigated pages by the Platform server. Some of such scripts allow to dynamically select the displayed components. Thus, the logical description of only the selected components is considered for adaptation to the target device.

4.6.2.3.2. Copying and sharing

Three Italian researchers, Fabio, Giuseppe and Carmen are involved in a work meeting in London. At the end of the meeting, in the afternoon, they have to go back to Italy. Fabio navigates the British Airways Web site through the proxy of the Platform and performs some searches. He finds the best solution for flying from London Gatwick to Pisa in the late evening. Before paying the ticket with his credit card, Fabio sends the page with the chosen travel itinerary, date and preferences towards Giuseppe and Carmen's laptops. A copy of the partially completed form is then available to Carmen and Giuseppe, who can quickly proceed to pay their tickets with their credit cards, without having to fill the form of the itinerary (since it has already been completed by Fabio), and be sure to get the same flight as their colleague.

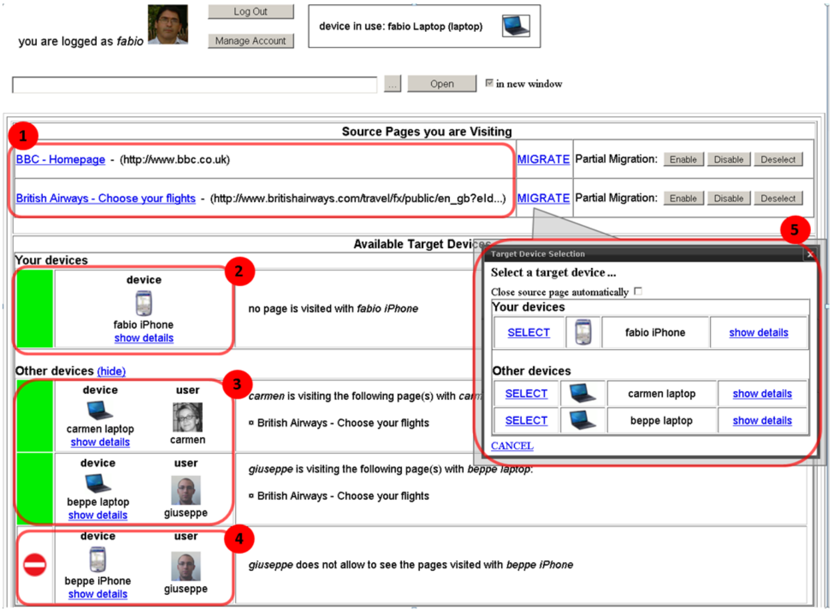

Figure 50. The Multi-Device Control Panel in the scenario considered.

Figure 50 shows the situation, on the Multi-device Control Panel of one user (Fabio), after his interface has been shared. In the "Available Target Devices" section, among the information about the devices belonging to other users, there is the current state of the navigation activity (i.e. Carmen and Giuseppe are confirming their ticket in the "British Airways – Choose Your Flight" page). In detail, with respect to Figure 50, the red-highlighted parts are indexed as follows:

- *(1)* title and URL of the pages visited by the user through the current device (Fabio Laptop);
- *(2)* other devices owned by the same user and, thus, available to him as targets;
- *(3)* devices of other users that are selectable as targets by Fabio;
- *(4)* a device that is not selectable as target by Fabio (and for which the visited pages are not visible);
- *(5)* menu for selecting the target device.

The “Target Device Selection” menu, that pops-up when clicking “MIGRATE”, is the dialog for selecting the destination device and for specifying whether the source page should be automatically closed. The destination device will receive a copy of the source interface (or of the selected parts, in case of partial migration).
Note that “beppe iPhone” device, which cannot be selected as a target for migration by Fabio, does not appear in the target devices list.

4.7 Final remarks

This chapter has tackled migratory Web user interfaces. Since this dissertation copes with user interfaces for mobile devices, particular focus was given on partial migration of Web pages. Partial migration, indeed, allows to reduce spatial complexity of the user interface, so as to allow better fitting for small displays.

The preliminary method is based on the use of user interface description languages. An automatic reverse engineering of the user interface builds specification of UIs compliant with such languages, starting from existing Web content (i.e. the desktop version of the user interface). The reverse engineering phase is able to handle all the (X)HTML and CSS tags. In this phase, it is possible to identify the structure of the set of elements populating the page, which is a crucial point for partial migration, in order to be able to present the user with the set of elements that can be migrated. Among the benefits of this strategy, there is the simplification of the page structure that presents the elements. Indeed, the tree-structure of the page is obtained through the reverse engineering of the document. The procedure actually performs an abstraction of the page, thus the tree-structure is populated with elements that refer to the most relevant HTML components. The main limitations of this approach lie both in the meaningfulness of the tree structure and in the efficiency for populating it and for carrying out the elements selection. The low meaningfulness is due to the lack of an intuitive labeling mechanism for the tree elements (i.e. the user does not immediately grasp the correspondence between the tree nodes and the page components). The low efficiency is related to the general Platform architecture. In particular, the solution presented was based on a Migration Client implemented as a stand-alone application, which communicated to the client browser indirectly (through the Migration Server). This indirect communication, together with the need for performing a reverse engineering of the whole page before obtaining the tree structure, were the main responsible for the latency issues.

Some of the previously mentioned limitations were solved with the implementation of a totally Web based Migration Client. On the one hand, the inter-window communication between the Migration Client and the navigated Web page reduced the latency of the components selection. On the other hand, the direct selection of components on the navigated page through mouse hovering/clicking did not require a preliminary reverse engineering of the Web page (since the selection was performed on the original HTML tree structure).

The improved architecture of the Migration Platform and the Web-based Migration Client have enabled migration in multi-device as well as in multi-user contexts.

A mechanism for managing privacy policies has been presented, also reporting on a user test. In general, the Migration Platform is able to migrate the Web pages through both push and pull modality without posing any constraint on their authoring and without requiring any manual modification. In a few cases there are still issues in preserving all the state, for instance when the pages use specific libraries (e.g. JQuery) to dynamically modify the DOM.
The user test carried out has led to interesting and promising indications on the general reaction of users with respect to usability and usefulness of the proposed platform. The test also allowed to identify the parts that need further consideration, even with respect to more technical indicators, such as time performance.

The client-server support of the Migration Platform has been analyzed and discussed with respect to security issues posed by multi-device and multi-user migration of interactive Web applications. The discussion has also reported on how security strategies have been applied to the existing Migration Platform, which did not address them previously. The results have been validated by testing various widely known Web applications to ensure that they effectively migrate with state persistence when the security support is applied (i.e. when migration is carried out over HTTPS).

A more general discussion was finally reported about multi-device access. Indeed, usability of pervasive systems supporting it is still a challenge. The discussion has tackled the main dimensions that characterize user interfaces able to support the multi-device access. A framework referring to those dimensions has been introduced and its relationships with the Migration Platform have been highlighted. It has been outlined that the Migration Platform covers most dimensions of the framework. The few points in the framework that are not covered can be the topic for new research work. For instance, since the Platform is based on a migration/proxy server, it would be interesting to investigate solutions able to exploit peer-to-peer communication among sets of devices that are opportunistically accessed. Another point is that the Platform supports interoperability by exploiting model-based languages, but currently focuses on Web applications. Thus, it would be interesting to involve other types of applications (e.g. Java, Silverlight, etc.) in the multi-device support. In this context, further in-depth investigation on how the task type and application domains impact on the dimensions considered would be needed. Users' attitudes towards the multi-device and multi-user platform when the context is partially or totally shared with others, especially when performing tasks with privacy concerns, is another topic that deserves further study. In addition, support of richer set of interaction modalities and their various possible combinations seems an area that needs to be better explored considering also the technological improvements for supporting modalities such as voice and gesture.

5. CONCLUSIONS AND FUTURE PERSPECTIVES

The main subject of this dissertation has been an investigation about development of user interfaces for mobile devices and techniques for user interface migration, with particular focus on mobile target devices.
The user study of the initial prototype of mobile museum guide has revealed a good interest of the users involved in the trial. Despite the huge number of additional features, such as location awareness, educational games, multi-device capabilities, the users (10 out of 12) have preferred the enhanced version of the mobile guide. The only two choices for the basic version were driven by "speed" issues (i.e. the need to perform a quick museum visit) and by the user need to have more freedom to take initiative.
This result indicates that it is worth continuing to enhance the features, though additional work is needed in order to maximize usability. In particular, it was declared that it is not always easy to follow a description that is partially displayed in multiple screens (i.e. PDA and large display). This observation highlights that further investigation is needed with regard to distributed user interfaces in museum environments.
With respect to the multimodality, six users preferred the vocal part, one the graphical version and three the vocal plus graphic version (while two did not report any preference). Thus, according to such results, the vocal output is a reasonable alternative or a complement to the traditional graphical one.
The games interfaces were rated quite well, and the users found them useful for stimulating learning as long as the games are easy to solve. Some participants suggested to dynamically enable only the games related to visited areas, thus confirming their interest in the edutainment dimension of the solution.
Improvements on the localisation performance were strongly recommended, due to the delays in responsiveness.
Users' attitude to follow the system suggestions was studied through a log analysis, revealing that only about one third of the suggestions were actually followed. This indicates that most participants preferred to visit museums autonomously, although the mobile guide was considered in general useful and with good potential.
The above cited findings can serve as a starting point for future work, which might be dedicated to further empirical validation, aiming to collect both qualitative and quantitative data from a large set of museum visitors. New ways to enhance personalisation of the user interface, in particular in group visits, can be identified.

The case study of the museum guide has also considered accessibility aspects, i.e. mechanisms that enable impaired people to access the capabilities of the guide. Some of the solutions implemented for the educational museum guide have been indeed reused and improved for providing accessibility to the guide. A version for visually impaired visitors has been obtained by enhancing the location awareness with direction detection support and by integrating the multimodal user interface with vibrotactile feedback. The aim was to facilitate orientation among tagged objects and obstacle avoidance.

An iterative design process has led to a final prototype, which result on a quite complete and unobtrusive solution, based on the use of vibrotactile feedback in a novel manner.
Usability test and statistical analyses have been performed. The main findings indicate that the users appreciate vocal synthesis with respect to traditional audio clips. Vibrotactile feedback, particularly when combined with vocal sentences, is considered to add value to the multimodal solution. Vibrotactile feedback is perceived as unobtrusive and non-annoying especially for repetitive notifications, such as the presence of frequent obstacles. In this regard, the involved users did not express a net preference towards particular … of the obstacle avoidance feedback: while some preferred a continuous feedback proportional to the obstacle distance and considered it as “safer”, others found as less annoying the feedback driven by obstacle approaching (i.e. provided only when the obstacle distance decreases).
From the user study, it has been observed that different subjects have in general different preferences: some of them need intense vibration in order to detect the feedback, while others suffer “pins and needles” sensations or other diseases and thus require less intense vibration. Thus, despite the validity of the vibrotactile feedback to convey certain type of information, it can be concluded that the haptic output should be customizable to let the user choose the preferred mode (e.g., continuous or approach-dependent vibration to indicate obstacle proximity).
Possible future improvements of the presented prototype could even include context-dependent feature, such as the signalling of obstacles (e.g., other people) through continuous vibration with automatic switch to the discontinuous mode when the target item is reached.
Other improvements in the localization capabilities can be achieved by considering different, complementary or alternative technologies. Further modalities for conveying information could be investigated as well, such as temperature and electricity.

Some of the techniques previously experimented have been employed on a different domain. The localization and orientation supports, that enabled mobility on the accessible museum guide, were reused on a different experimental tool for mobile devices. Although the tool was also aimed to virtually represent a visited environment on a mobile device, the final aim was different from the museum guide. The tool, referred to as Device Selection Map, was devoted to facilitate the selection of target stationary devices for task migration.
The outlines of the user study reveal a preference towards the map-centered display mode for the virtual environment representation, with respect to the user-centered one. This result seems in contrast to what was expected, since the latter modality is widely used in everyday applications (such as navigation systems). Thus, additional investigation would be needed in order to further motivate the user’s attitude that emerged from the discussed study.

Migratory user interfaces, with particular focus on desktop to mobile device migration, were considered in the last chapter of this dissertation. Several aspects have been tackled. The study has found that, due to the heterogeneity of devices involved in migration, it is worth to implement the user control panel (Migration Client) as a Web application, in order to provide flexibility and compatibility. Both

partial and total Web page migration techniques have been evaluated through user studies, confirming that the usability of the Web Migration Client is better than the stand-alone version.
The automatic adaptation, driven by a reverse engineering approach that exploits model-based languages, has shown to preserve quite well content and functionalities of the migrated/adapted Web pages. Further future work in this direction could consider types of devices/modalities other than desktop and mobile devices, for example migration from desktop to vocal devices.

Security of the Migration Platform has been investigated too. According to static and run-time analyses, the enhanced Platform guarantees security and privacy. In this regard, future work is needed to evaluate the usability of the security mechanisms.

The Migration Platform has be finally studied within a multi-dimensional framework for multi-device application access. The study revealed which dimensions of the framework are covered by the Platform and how. The few points in the framework that are still not covered can be the topic for new research work. The discussed dimensions have also highlighted possible developments for the Platform implementation, such as the conversion from a server-based to a peer-to-peer architecture.

A deeper technical evaluation about scalability is also highly desirable, as the Migration Platform is enriched with multi-user and social awareness capabilities. In this regard, performance and efficiency are relevant when large groups of users access the Platform at the same time.

REFERENCES

1. Abowd D.A., Atkeson, C.G., Hong, J., Long, S., Pinkerton, M., "Cyberguide: a mobile context-aware tour guide", *Wireless Networks* 3 (5), 1996, pp. 421–433.

2. Alapetite A., "Dynamic 2D-barcodes for multi-device Web session migration including mobile phones", Personal and Ubiquitous Computing, Springer-Verlag London, 14 (1), 2010.

3. Amemiya T., Sugiyama H., "Design of a haptic direction indicator for visually impaired people in emergency situations", In: 11th International Conference on Computers Helping People with Special Needs (ICCHP), Linz, Austria, Lecture Notes in Computer Science, vol. 5105, 2008, pp. 1141–1144.

4. Balme L., Demeure A., Barralon N., Coutaz J., Calvary G., "CAMELEON-RT: A Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces", In: Proceedings of EUSAI 2004, LNCS, Springer-Verlag, 2004, pp. 291-302.

5. Banâtre M., Couderc P., Pauty J., Becus M., "Ubibus: ubiquitous computing to help blind people in public transport", In: MobileHCI 2004, Lecture Notes in Computer Science, vol.3160, 2004, pp.310–314.

6. Bandelloni R., Mori G., Paternò F., "Dynamic Generation of Migratory Interfaces", In: MobileHCI 2005, Salzburg, Austria, ACM Press, 2005, pp. 83-90.

7. Bandelloni R., Paternò F., "Flexible Interface Migration", In: Proceedings of IUI 2004, ACM Press, 2004, pp.148-157.

8. Baudisch P., Irani P., Gustafson S., Gutwin C., "Wedge: Clutter-Free Visualization of Off-Screen Locations", In: Proceedings of CHI 2008, pp. 787-796.

9. Baudisch P., Rosenholtz R., "Halo: A technique for visualizing off-screen locations", In: Proceedings of CHI 2003, pp. 481–488.

10. Baudisch P., Xie X., Wang C., Ma W.-Y., "Collapse-to-Zoom: Viewing Web Pages on Small Screen Devices by Interactively Removing Irrelevant Content", In: Proceedings of UIST 2004, ACM New York, 2004, pp. 91-94.

11. Bell M., Chalmers M., Barkhuus L., Hall M., Sherwood S., Tennent P., Brown B., Rowland D., Benford S., Capra M., Hampshire A., "Interweaving mobile games with everyday life", In: Proceedings of CHI 2006, Games and Performances, 2006, Montréal, Canada.

12. Bellotti F., Berta R., De Gloria A., Margarone M., “Guiding Visually Impaired People in the Exhibition”, In: Proceedings of Mobile Guide ‘06, 2006, Turin, Italy.

13. Bellotti F., Berta R., De Gloria A., Ferretti E., Margarone M., “VeGame: exploring art and history in Venice”, *Computer* 36 (9), 2003, pp. 48–55.

14. Bellucci F., Ghiani G., Paternò F., Porta C., “Automatic Reverse Engineering of Interactive Dynamic Web Applications to Support Adaptation across Platforms”, In: Proceedings of IUI 2012, February 14-17, 2012, Lisbon, Portugal.

15. Bellucci F., Ghiani G., Paternò F., Santoro, C., “Engineering JavaScript State Persistence of Web Applications Migrating across Multiple Devices”, In: Proceedings of EICS 2011, ACM, 2011, pp. 105-110.

16. Benedek J., Miner T., Measuring Desirability: New methods for evaluating desirability in a usability lab setting. In: Proceedings of Usability Professionals' Association Conference, July 8-12, 2002, Orlando, Florida, USA.

17. Bila N., Ronda T., Mohomed I., Truong K.N., de Lara E., “PageTailor: reusable end-user customization for the mobile Web”, In: Proceedings of MobiSys 2007, ACM, 2007, pp. 16-29.

18. Blom J., Mäntylä M., Oulasvirta A., Rantanen M., Tiitta S., “InfoRadar: group and public messaging in the mobile context”, In: Proceedings of NordiCHI 2004, ACM Press, 2004, pp. 131-140.

19. Blumendorf M., Roscher D., Albayrak S., “Dynamic User Interface Distribution for Flexible Multimodal Interaction”, In: Proceedings of ICMI-MLMI’10, ACM, 2010, pp. 20:1-20:8.

20. Brewster S.A., Chohan F., Brown L.M., “Tactile feedback for mobile interactions”, In: CHI2007, San Jose, CA, USA, ACM Press, Addison-Wesley, pp. 159–162.

21. Brignull H., Rogers Y., “Enticing People to Interact with Large Public Displays in Public Spaces”, In: Proceedings of Interact 2003, Zurich, Switzerland, pp. 17–24.

22. Brooke J., “SUS - A quick and dirty usability scale”, http://usabilitynet.net/trump/documents/Suschapt.doc (1986).

23. Brown B., MacColl I., Chalmers M., Galani A., Randell C., Steed A., “Lessons from the lighthouse: collaboration in a shared mixed reality system”, In: Proceedings of CHI 2003, ACM, New York, pp. 577–584.

24. Brown L. M., Brewster S.A., Purchase H.C., “Multidimensional tactons for non-visual information display in mobile devices”, In: MobileHCI 2006, Espoo, Finland, ACM, pp. 231–238.

25. Burigat S., Chittaro L., Gabrielli S., “Visualizing locations of off-screen objects on mobile devices: A comparative evaluation of three approaches”, In: Proceedings of MobileHCI 2006, pp. 239-246.

26. Ceipidor U.B., Medaglia C.M., Mei M., Sabene M., Serbanati A., “GLIDEO: a RFID system for identification and description of objects by blind and visually impaired people”, In: Proceedings of EURFID Forum 2007, Brussels, Belgium.

27. César P., Vaishnavi I., Kernchen R., Meissner S., Hesselman C., Boussard M., Spedalieri A., Bulterman D., Gao B., “Multimedia adaptation in ubiquitous environments: benefits of structured multimedia documents”, In: Proceedings of the eight ACM Symposium on Document Engineering (DocEng ’08), ACM New York, 2008, pp. 275-284.

28. Chang T.H., Li Y., “Deep Shot: A Framework for Migrating Tasks Across Devices Using Mobile Phone Cameras”, In: Proceedings of CHI 2011, ACM, 2011, pp. 2163-2172.

29. Cheverst K., Davies N., Mitchel K., Smyth P., “Providing tailored context-aware information to city visitors”, Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2000), vol. 1892. Springer-Verlag, Berlin Heidelberg/New York, pp. 73–85 (LNCS).

30. Chumkamon S., Tuvaphanthaphiphat P., Keeratiwintakorn P., “A blind navigation system using RFID for indoor environments”, In: 5th International Conference on Electrical/Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON2008), 2008, Krabi, Thailand, pp. 765–768.

31. Ciavarella C., Paternò F., “The design of a handheld, location-aware guide for indoor environments”, *Personal Ubiquitous Computing* 8 (2), May 2004, pp. 82-91.

32. Coroama V., “Experiences from the design of a ubiquitous computing system for the blind”, In: CHI 2006 Extended Abstracts on Human Factors in Computing Systems, pp. 664-669

33. Cosley D., Lewenstein J., Herman A., Holloway J., Baxter J., Nomura S., Boehner K., Gay G., “ArtLinks: fostering social awareness and reflection in museums”, In: Proceedings of CHI 2008, ACM, pp. 403-412.

34. Cross-Origin Resource Sharing. W3C Working Draft 27 July 2010. http://www.w3.org/TR/cors/.

35. D’Atri E., Medaglia C.M., Serbanati A., Ceipidor U.B., Panizzi E., D’Atri A., “A system to aid blind people in the mobility: a usability test and its results”, In: Second International Conference on Systems (ICONS’07), IEEE, 2007, pp. 35-40.

36. Daimon T., Kawashima H., Mashimo S., "Driver's Characteristics for Map Information Representation (North up Map/Heading up map) in Navigation Displays", In: Proceedings of VNIS '93, IEEE, 1993, pp. 410-413.

37. Dearman D., Pierce J., "It's on my Other Computer!:, Computing with Multiple Devices", In: Proceedings of CHI 2008, ACM Press, 2008, pp. 767-776.

38. Demeure A., Sottet J.-S., Calvary G., Coutaz J., Ganneau V., Vanderdonckt J., "The 4C Reference Model for Distributed User Interfaces", In: Proceedings of ICAS '08, IEEE, 2008, pp. 61-69.

39. Denis C., Karsenty L., "Inter-usability of multidevice systems – a conceptual framework", In: Seffah A. and Javahery H. (eds.) Multiple User Interfaces: Cross-Platform Applications and Context-Aware Interfaces. Wiley & Sons, 2005.

40. Ding Y., Huber J., "Designing multi-user multi-device systems: an architecture for multi-browsing applications", In: Proceedings of MUM '08, ACM Press, 2008, pp. 8-14.

41. Dini R., Paternò F., Santoro C., "An Environment to Support Multi-user Interaction and Cooperation for Improving Museum Visits through Games", In: Mobile HCI 2007, ACM Press, pp. 515-521.

42. Edwards W.K., Newman M.W., Sedivy J.Z., Smith T.S., "Experiences with Recombinant Computing: Exploring Ad Hoc Interoperability in Evolving Digital Networks", ACM Transactions on Computer-Human Interaction, 16 (1), Article 3, 2009.

43. Eisenstein J., Vanderdonckt J., Puerta A.R., "Applying model-based techniques to the development of UIs for mobile computers", In: Proceedings IUI 2001, ACM New York, 2001, pp. 69-76.

44. Fischer C., Gellersen H., Gostner R., Guinard D., Kortuem G., Kray C., Rukzio E., Strengin S., "Supporting Device Discovery and Spontaneous Interaction with Spatial References", *Personal and Ubiquitous Computing*, 13 (4), 2009.

45. Fonseca J.M.C. (ed.). W3C Model-Based UI XG Final Report, May 2010, http://www.w3.org/2005/Incubator/model-based-ui/XGR-mbui-20100504/.

46. Fraser M., Stanton D., Hui Ng K., Benford S., O'Malley C., Bowers J., Taxen G., Ferris K., Hindmarsh J., "Assembling History: Achieving Coherent Experiences with Diverse Technologies", In: ECSCW 2003, Kluwer, pp. 179-198.

47. Ghiani G., Leporini B., Paterno F., "Supporting orientation for blind people using museum guides", In: CHI 2008 Extended Abstracts, ACM Press, 2008, pp.3417-3422.

48. Ghiani G., Leporini B., Paterno F., “Vibrotactile feedback as an orientation aid for blind users of mobile guides”, in: MobileHCI 2008, Amsterdam, The Netherlands, ACM Press, 2008, pp. 431-434.

49. Ghiani G., Paternò F., Santoro C., Spano L.D., “A location-aware guide based on active RFIDs in multi-device environments”, In: Proceedings of CADUI 2008, Springer-Verlag, pp. 59-70.

50. Grinter R., Aoki P., Hurst A., Szymanski M., Thornton J., Woodruff A., “Revisiting the visit: understanding how technology can shape the museum visit”, In: Proceedingss of CSCW’02, ACM Press, 2002, pp. 146-155.

51. Grolaux D., Van Roy P., Vanderdonckt J., “Migratable User Interfaces: Beyond Migratory Interfaces”, In: Proceedings of First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04), IEEE, 2004, pp. 422-430.

52. Han R., Perret V., Naghshineh M., “WebSplitter: a unified XML framework for multi-device collaborative Web browsing”, In: Proceedings of CSCW 2000, ACM, 2000, pp. 221-230.

53. Hatala M., Wakkary R., “Ontology-based user modeling in an augmented audio reality system for museums”, *User Modeling and User-Adapted Interaction* 15 (3), 2005, pp. 339-380.

54. Hermann F., Bieber G., Duesterhoeft A., “Egocentric maps on mobile devices”, In: Proceedings of 4th International Workshop on Mobile Computing, IMC 2003, Verlag, 2003, pp. 32-37.

55. Hermann F., Schmidt-Belz B., “User Validation of a Nomadic Exhibition Guide”, In: Proceedings of MobileHCI 2004, Springer LNCS, Vol. 3160, 2004, pp.86-97.

56. Hindmarsh J., Heath C., Vom Lehn D., Ciolfi L., Hall T., Bannon L., “Social Interactions in Museums and Galleries”, Deliverable of the Shape Project (D2.1), 2001, Disappearing Computer Project.

57. Hoggan E., Crossan A., Brewster S.A., Kaaresoja T, “Audio or Tactile Feedback: Which Modality When?”, In: Proceedings of CHI 2009, ACM, 2009, pp. 2253-2256.

58. HttpClient Overview, http://hc.apache.org/httpcomponents-client-ga/.

59. Jimenez Pazmino P., Lyons L., “An exploratory study of input modalities for mobile devices used with museum exhibits”, In: Proceedings of CHI 2011, ACM, 2011, pp. 895-904.

60. Johanson B., Ponnekanti S., Sengupta C., Fox A., “Multibrowsing: Moving Web Content across Multiple Displays”, In: Proceedings of Ubicomp 2001, Springer-Verlag, 2001, LNCS 2201, pp. 346-353.

61. Kildal J., Brewster S.A., "EMA-tactons: vibrotactile external memory aids in an auditory display", In: Proceedingss of the IFIP Interact 2007, Rio de Janeiro, Brazil, Lecture Notes in Computer Science, vol. 4663, Springer, pp. 71-84.

62. Kozuch M., Satyanarayanan M., "Internet Suspend/Resume", In: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'02), IEEE Press, 2002.

63. Kray C., Laakso K., Elting C., Coors V., "Presenting route instructions on mobile devices", In: Proceedings of IUI 2003, ACM Press, 2003, pp. 117-124.

64. Leikas J., Stromberg H., Ikonen V., Suomela R., Heinila J., "Multi-user Mobile Application and a Public Display: Novel Ways for Social Interaction", In: PerCom 2006, IEEE Press, pp. 66-70.

65. Leinhardt G., Crowley K., "Objects of learning, objects of talk: changing minds in museums", In: Paris, S. (Ed.), *Multiple Perspectives on Object Centered Learning*, Lawrence Erlbaum, Hillsdale, NJ, 2002, pp. 301-324.

66. Lima A.C.O., Silverio Freire R.C., De Assis F.M., Silva e Silva R.C., Ramos de Souza A.F., "Comparative analysis of tactile sensibility between the fingers in vibro-tactile stimulus recognition", In: Instrumentation and Measurement Technology Conference, IEEE, 2007, pp. 1-6.

67. Liu A.X., Kovacs J., Huang C.-T., Gouda M.G., "A Secure Cookie Protocol for HTTP", In: Proceedings of ICCCN 2005, IEEE, 2005, pp. 333-338.

68. Lorenz A., Fernandez De Castro C., Rukzio E., "Using handheld devices for mobile interaction with displays in home environments", In: Proceedings of MobileHCI 2009, ACM New York, 2009.

69. Luyten K., Coninx K., "Distributed User Interface Elements to support Smart Interaction Spaces", In: Proceedings of IEEE Symposium on Multimedia, Irvine, California, 12-14 December 2005, pp. 277-286.

70. Manduchi R., Coughlan J., "Portable and mobile systems in assistive technology: introduction to the special thematic session", In: 11th International Conference on Computers Helping People with Special Needs (ICCHP), Linz, Austria, Lecture Notes in Computer Science, vol. 5105, 2008, pp. 1078-1080.

71. Mäntyjärvi J., Paternò F., Salvador Z., Santoro C., "Scan and Tilt: Towards Natural Interaction for Mobile Museum Guides", In: Proceedings of MobileHCI 2006, ACM Press. pp. 191-194.

72. Maurer M., De Luca A., Stockinger, T., "Shining Chrome: Using Web Browser Personas to Enhance SSL Certificate Visualization", In: Proceedings of INTERACT 2011, Springer-Verlag, 2011, Vol. 4, pp. 44-51.

73. Melchior J., Grolaux D.,Vanderdonckt J., Van Roy P., "A Toolkit for Peer-to-Peer Distributed User Interfaces: Concepts, Implementation, and Applications", In: Proceedings of EICS 2009, ACM New York, 2009, pp. 69-78.

74. Moreno L., Galvez M.C., Ruiz B., Martinez P., "Inclusion of accessibility requirements in the design of electronic guides for museums", In: 11th International Conference on Computers Helping People with Special Needs (ICCHP), Linz, Austria, Lecture Notes in Computer Science, vol. 5105, Springer, Heidelberg, 2008, pp. 1101-1108.

75. Myers B.A., "Using handhelds and PCs together", **Communications of the ACM**, 44 (11), ACM, 2001, pp. 34-41.

76. Nacenta M., Aliakseyeu D., Subramanian S., Gutwin C.A., "Comparison of Techniques for Multi-Display Reaching", In: Proceedings of CHI 2005, ACM Press, 2005, pp. 371-380.

77. Newman M.W., Izadi S., Edwards W.K., Sedivy J.Z., Smith T.F., "User Interfaces When and Where They are Needed: An Infrastructure for Recombinant Computing", In: Proceedings of UIST 2002, ACM, pp. 88-100.

78. Nichols J., Hua Z., Barton J., "Highlight: a System for Creating and Deploying Mobile Web applications", In: Proceedings of UIST 2008, ACM, 2006, pp. 249-258.

79. Nichols J., Myers B.A., Higgins M., Hughes J., Harris T.K., Rosenfeld R., Litwack K., "Personal universal controllers: controlling complex appliances with GUIs and speech", In: Proceedings of CHI 2003, ACM New York, 2003, pp. 624-625.

80. Nichols J., Rothrock B., Chau D.H., Myers B.A., "Huddle: automatically generating interfaces for systems of multiple connected appliances", In: Proceedings of UIST 2006, ACM New York, 2006, pp. 279-288.

81. Oakley I., Kim Y., Lee J., Ryu J., "Determining the feasibility of forearm mounted vibrotactile displays", In: Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, IEEE, 2006, p. 81.

82. Olsen D.R., Nielsen S.T., Parslow D., "Join and Capture: a Model for Nomadic Interaction", In: Proceedings of UIST '01, ACM, 2001, pp. 131-140.

83. Opperman R., Specht M., Jaceniak I., "Hippie: A Nomadic Information System. Handheld and Ubiquitous Computing", Springer-Verlag, Berlin, 1999, pp. 330-333.

84. Paternò F., Santoro C., Scorcia A., "Ambient Intelligence for Supporting Task Continuity across Multiple Devices and Implementation Languages", *The Computer Journal*, Oxford University Press, 53 (8), 2010.

85. Paternò F., Santoro C., Spano L.D., "MARIA: A Universal Language for Service-Oriented Applications in Ubiquitous Environment", *ACM Transactions on Computer-Human Interaction*, 16 (4), ACM New York, November 2009, pp.19:1-19:30.

86. Petrelli D., Not E., “User-centered design of flexible hypermedia for a mobile guide: reflections on the HyperAudio experience”, *User Modeling and User-Adapted Interaction* 15 (3–4), 2005, pp. 303-338.

87. Petrelli D., Not E., Sarini M., Stock O., Strapparava C., Zancanaro M., “HyperAudio: location-awareness + adaptivity”, In: Proceedingss of CHI ’99 Extended Abstracts, ACM Press, 1999, pp. 21-22.

88. Pierce J.S., Nichols J., “An infrastructure for extending applications' user experiences across multiple personal devices. In: Proceedings of UIST 2008, ACM New York, pp. 101-110.

89. Poslad S., Laamanen H., Malaka R., Nick A., Buckle P., Zipf A., “CRUMPET: Creation of User-Friendly Mobile Services Personalised for Tourism”, *3G Mobile Communication Technologies*, London, UK, 2001, pp. 26-29.

90. Pospischil G., Umlauft M., Michlmayr E., „Designing LoL@, a Mobile Tourist Guide for UMTS”, *Information Technology and Tourism*, vol. 5 (3), 2003, Cognizant Communication Corporation, pp. 151-164.

91. Puikkonen A., Sarjanoja A., Haveri M., Huhtala J., Häkkilä J., “Towards designing better maps for indoor navigation: experiences from a case study”, In: Proceedings of 8th international Conference on Mobile and Ubiquitous Multimedia (MUM '09), ACM, 2009, pp. 1-4.

92. Quan D., Huynh D., Karger D.R., Miller R., “User interface Continuations”, In: Proceedings of UIST ‘03. ACM, 2003, pp. 145-148.

93. Rekimoto J., “Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments”, In: Proceedings of UIST ‘97, pp. 31-39, 1997.

94. Seager W., Fraser D.S., “Comparing Physical, Automatic and Manual Map Rotation for Pedestrian Navigation”, In: Proceedings of CHI 2007, ACM Press, 2007, pp. 767-776.

95. Schmidt A., “Implicit Human Computer Interaction Through Context”, In: **Personal and Ubiquitous Computing** 4(2/3), Springer, 2000, pp. 191-199.

96. Shah C., Bouzit M., Youssef M., Vasquez L., “Evaluation of RUN-etra - tactile feedback navigation system for the visually-impaired”, In: The International Workshop on Virtual Rehabilitation, 2006, pp. 72-77.

97. Sharp R., Madhavapeddy A., Want R., Pering T., “Enhancing Web browsing security on public terminals using mobile composition”, In: Proceedings of MobiSys 2008, ACM, 2008, pp. 94-105.

98. Sharp R., Scott J., Beresford A.R., “Secure Mobile Computing via Public Terminals”, In: Proceedings of Pervasive 2006, Springer-Verlag, 2006, pp. 238-25.

99. Siegel S., Castillan N.J., **Nonparametric Statistics for the Behavioral Sciences**, second ed., McGraw-Hill, New York, 1988, pp. 87-95, 174-183.

100. Sohn T., Li F.C.Y., Battestini A., Setlur V., Mori K., Horii H., “Myngle: unifying and filtering web content for unplanned access between multiple personal devices”, In: Proceedings of UbiComp 2011, ACM, pp. 257-266.

101. Sparacino F., “The museum wearable: real-time sensor-driven understanding of visitors’ interests for personalized visually-augmented museum experiences”, In: Proceedingss of Museums and the Web, 2002.

102. Stock O., Zancanaro M., Busetta P., Callaway C., Krüger A., Kruppa M., Kuflik T., Not E., Rocchi C., “Adaptive, intelligent presentation of information for the museum visitor in PEACH”, *User Modeling and User-Adapted Interaction* 17 (3), 2007, pp. 257-304.

103. Tomitsch M., Schlögl R., Grechenig T., Wimmer C., Költringer T., “Accessible real-world tagging through audio-tactile location markers”, In: Proceedings of NordiCHI 2008 Conference, October 20–22, 2008, Lund, Sweden, p. 551.

104. Tsetsos V., Anagnostopoulos C., Kikiras P., Hadjiefthymiades S., “Semantically enriched navigation for indoor environments”, *International Journal of Web and Grid Services*, Inderscience Publishers, 2 (4), 2006, pp. 453-478.

105. Vandervelpen C., Conix K., “Towards Model-Based Design Support for Distributed User Interfaces”, In: Proceedings of NordiCHI 2004, ACM, 2004, pp. 61-70.

106. Wäljas M., Segerståhl K., Väänänen-Vainio-Mattila K., Oinas-Kukkonen H., “Cross-Platform Service User Experience: A Field Study and an Initial Framework”, In: Proceedings of MobileHCI 2010, ACM, 2010, pp. 219-228.

107. WhiteHat security report https://www.whitehatsec.com/home/assets/WPstats_winter11_11th.pdf?doc=WPstats_winter11_11th

108. Willis S., Helal S., “RFID information grid for blind navigation and wayfinding”, In: 9th IEEE Symposium on Wearable Computers, IEEE, 2005, pp. 34-37.

109. Wiltse H., Nichols J., “PlayByPlay: collaborative Web browsing for desktop and mobile devices”, In: Proceedings of CHI 2009, ACM, 2009, pp. 1781-1790.

110. Zhao L., Iyer R., Makineni S., Bhuyan L., “Anatomy and Performance of SSL Processing”, In: Proceedings of ISPASS 2005, IEEE, 2005, pp.197-206.

LIST OF FIGURES

LIST OF TABLES