Automatic User Interface Generation and Migration in Multi-Device Environments

Renata Bandelloni, Giulio Mori, Fabio Paternò
ISTI-CNR
Via G.Moruzzi 1
56124 Pisa, Italy

Abstract

Ubiquitous computing systems provide for migratory user interfaces that allow users to move about freely while carrying on their tasks through a variety of interaction platforms. We present our environment, which provides runtime support to obtain migratory interfaces automatically generated for different platforms. The solution is based on a migration/proxy server, which is able to generate interfaces for the target device with the state resulting from previous user interactions in order to support platform adaptation and task performance continuity. Thus, it allows users to receive user interfaces adapted to their device even if only a desktop version was originally developed and, when they change device, they can continue from the point they left off on the previous device without having to enter again the information already provided.

Categories and Subject Descriptors: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Evaluation/methodology; Input devices and strategies; Interaction styles; D.2.2 [Software Engineering]: Design Tools and Techniques—User interfaces
General Terms: Design, Human Factors, Reliability
Additional Key Words and Phrases: Migratory interfaces, Adaptation to interaction platform, Model-based design, Multi-device interfaces.

Introduction

The availability of various types of interaction platforms and the possibility of freely changing devices without having to restart the session from scratch are important characteristics of ubiquitous environments. For this reason, recently there has been an increase of interest in migratory interfaces. They provide users with the possibility of changing the interaction device and still continue their tasks through an interface adapted to the new platform. Migration can involve devices belonging to different platforms. The concept of platform is used to group those devices that have similar interaction resources (such as the graphical desktop, the graphical PDA, the vocal device). Migratory interfaces require that the interface be able to adapt to the features of the new device in such a way as to satisfy usability criteria and support continuity in task performance.

In this paper, we present a solution able to support migration through several kinds of platforms with the possibility of automatically redesigning the existing application interface to adapt to their features. Our migration environment (GeReMi: Get Redesign Migrate) can involve different types of platforms offering different interaction modalities, such as graphical and vocal. The environment retrieves the original pages for the desktop from the Web and analyses them, thereby obtaining logical descriptions at different levels concerning both their user interface and the underlying task model. The logical descriptions are used to generate a redesigned page for the platform accessing the page or for the platform onto which the page is migrating. Since migration implies task performance continuity, the runtime interface state resulting from the previous user interactions is stored and adapted to the newly generated interface. Thus, our solution supports automatic generation of user interfaces for various platforms and even the possibility of dynamically migrating through them. This means that our environment allows developers to avoid developing multiple versions in a separate manner and limit themselves to implementing the desktop version, from which our runtime support generates the other interface versions as needed. Indeed, user interfaces for various platforms are generated from the Web desktop version independently of who implemented it and what authoring environment was used for this purpose assuming that standard W3C languages have been used.

In the paper, after a short discussion of related work, we first describe the architecture of our environment, we move on to explain how logical descriptions of the existing desktop version are created and then we indicate how such semantic information is used to redesign the interface for different modalities. An example application of the proposed approach is illustrated. We also report on user tests carried out on some applications of migratory interfaces. Lastly, conclusions and indications for future work are provided.

Related Work

The increasing availability of various types of interactive devices has raised interest in model-based approaches because they provide logical descriptions that can be used as a starting point for generating interfaces that adapt to the various devices at hand. In recent years, such interest has been accompanied by the use of XML-based languages to represent such logical descriptions (examples are XIML[1], UIML[2], TERESA-XML[3], USIXML[4]). However, such approaches have usually focused on providing support, not in the runtime phase, but only in the design and authoring phase in order to help designers to efficiently obtain different versions that adapt to the various interaction features.

Different techniques for redesigning Web applications for small devices have been proposed, see for example [5] [6] both orientated to obtaining a thumb-nailed view of the original page. These approaches are mainly based on a layout analysis and the small screen pages accessed through the thumbnails are built by extraction and summarisation of parts of the original page. In PUC [7] there has been a proposal aiming at dynamically generate user interfaces able to control a domestic appliance starting with its logical description, this work does not address the possibility of supporting users in continuing task performance when moving dynamically through different devices.

Supple [8] is an application and device independent system that automatically generates user interfaces for a variety of display devices. Supple uses decision-theoretic, combinatorial optimisation. Conceptually, Supple enumerates all possible ways of laying out the interface and chooses the one which minimizes the user's expected cost of interaction. The system uses a single level of functional specification language while our approach uses user interface description at three different abstraction levels for each different platform. Supple addresses exclusively graphical devices, and does not support interface migration.

ICrafter [9] is a solution to generate adaptive interfaces for accessing services in interactive spaces. It generates interfaces that adapt to different devices starting with XML-based descriptions of the service that must be supported. However, ICrafter is limited to creating support for controlling interactive workspaces by generating user interfaces for services obtained by dynamic composition of elementary ones and does not provide support for migration and, consequently, continuity of task performance across different devices.

Aura [10] provides support for migration but the solution adopted has a different granularity. In Aura for each possible application service various applications are available and the choice of the application depends on the interaction resources available. Thus, for example for word processing, if a desktop is available then an applications such as MS-Word can be activated, whereas in the case of a mobile platform a lighter editing application is used. Thus, Aura aims to provide a similar support but this is obtained mainly by changing the application depending on the resources available in the device in question, while we generate interfaces of the same application that adapt to the interaction resources available.

A general reference model for user interfaces aiming to support migration, distribution and adaptivity to the platform (termed *plasticity* in their article) is proposed in [11]. In our work we propose a concrete architecture, based on a client/server infrastructure, able to support migration of user interfaces associated with applications hosted by different content servers. While in our previous work [12] we found a solution based on the use of precomputed interfaces for different platforms that are dynamically activated, in this paper we present a solution, whereby, starting with the richest version (that is for the most powerful platform, the desktop), it is then possible to automatically generate versions for different platforms and associate them with the state resulting from the interactions performed on the previous devices. We moreover offer the possibility of migrating among devices with different interaction modalities. We introduced some preliminary ideas on how to obtain automatic generation of migratory Web interfaces in [13]. In this paper we are able to present a solution, along with an engineered prototype, supporting also migration through different modalities (such as voice) and report on first tests of the proposed tool and its results.

Software Architecture

The architecture of GeReMi is based on a migration/proxy server, which receives the access and migration requests and performs interface adaptation and interface runtime state management. The environment starts with pre-existing Web applications typically developed for desktop platform enhancing them with interface migration capabilities.

The users who want to access the service, have to load the migration client software onto their devices. This is a light weight piece of software used to communicate with the server to send migration requests, indicate when the device is available for migration and additional information related to the device, which can be useful in the migration process. It also permits to gather runtime information concerning the state of the interface running on the device, which results from the history of user interactions.

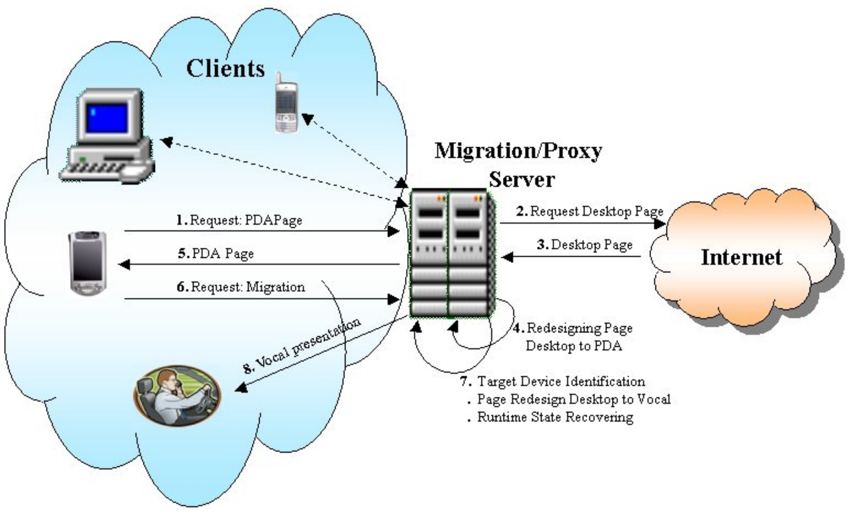


Figure 1: GeReMi Migration Service Architecture.

The migration/proxy server behaviour can be explained analysing the different situations it can cope with:

Simple proxy: The user accesses the Web through a device that belongs to the same platform type for which the pages were created for. The migration/proxy server retrieves the required page from the Web server and passes it on to the client.

Interface redesign. The user accesses the Web through a device that belongs to a platform different from that for which the pages were created, for example a vocal platform client accessing the pages designed for the desktop. First, the migration/proxy server retrieves the required page from the Web server, then the page is redesigned to be suitable for vocal interaction and the resulting interface is activated on the vocal device.

Migration from Desktop to non-Desktop platform. The user accesses the Web through a device whose platform is the same for which the pages were created and at a certain point migration towards a different platform is required. Let us consider for example a desktop to PDA migration. First, the target device of the migration has to be identified. Thus, the page that was accessed through the source device is stored and automatically redesigned for the PDA. This is enough to support platform adaptation, but there is still one step to be performed in order to preserve task performance continuity. The runtime state of the migrating page must be adapted to the redesigned page. In addition, the server has to identify the page to activate in the target PDA; this page is the one supporting the last task performed by the user, before migration. At the end of the process, the adapted selected page is sent to the PDA from which the user can continue the activity which was left off on the desktop.

Migration from non-Desktop platform to non-Desktop platform. The user accesses the Web through a device whose platform is different from the one for which the pages were created, and at a certain point migration towards another different platform is required. This is the most complex and interesting case, in which all the functionalities of our migration/proxy server are involved. This case differs from the previous one because both the source and the target device of migration do not match the platform of the original Web pages, and both of them require that the associated interfaces be redesigned. When redesigning the interface for the target device, the migration/proxy server exploits the logical description concerning the original platform interface already built. This is obtained by a double mapping first from the source redesigned interface to the original desktop interface, then from the original desktop interface to the target redesigned interface. As an example we can consider an application developed for desktop, accessed through a PDA and migrating to a mobile vocal phone. The steps to perform this migration example shown in Figure 1 are:

1. The user loads a Web page from a PDA
2. The server asks the content server for the original desktop page
3. The page is retrieved and stored in the migration/proxy server

4. The desktop page is redesigned for PDA going through the generation of the necessary logical descriptions
5. The redesigned page is sent to the PDA.
6. The user sends a migration request toward a vocal platform
7. The server recognizes and identifies the target platform and redesigns the interface for the vocal platform. Thus, even the runtime state of the migrating page is adapted to the newly interface generated.
8. The vocal interface containing the adapted state is sent to the target vocal platform.

Figure 2 provides more detail on how the server supports the migration. Through the use of reverse engineering techniques our environment builds the logical descriptions associated with the existing desktop version. When a request of migration arrives, the tool transforms the desktop logical descriptions into a version suitable for the target platform taking into account the modalities and interaction resources available. In addition, in this process the state resulting from previous user interactions (for example, text entered, elements selected and so on) is associated with the new interface version generated so that the user has not to enter such information again.

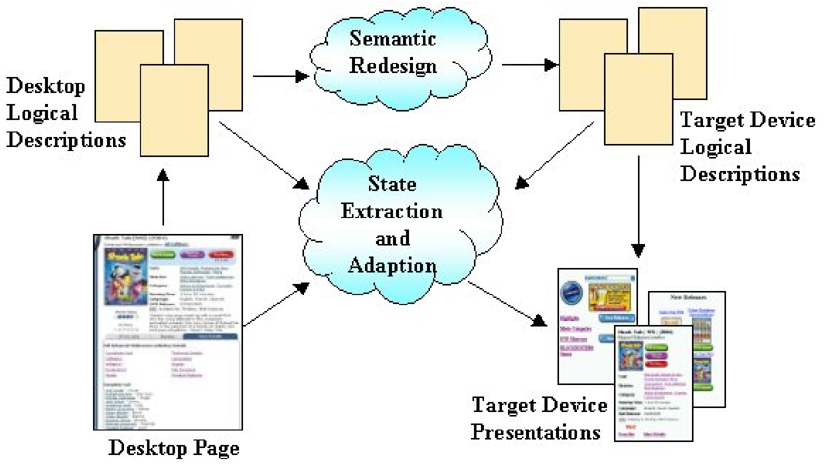

Semantic Redesign
Desktop Logical Descriptions
Target Device Logical Descriptions
State Extraction and Adaption
Desktop Page
Target Device Presentations

Figure 2. Processing performed on the server side for supporting migration.

Generating User Interface Logical Descriptions.

The generation of the user interface for the new platform starts with a reverse engineering step. From the Web page considered, three logical descriptions at different abstraction level are built: the concrete interface level, consisting in a platform-dependent description of the user interface but independent of the implementation language; the abstract interface level, consisting in a platform-independent description of the user interface; the task level, where the logical activities are considered.

There have already been proposals aiming to provide some support for reverse engineering of user interfaces. For example, WebRevEnge [14] automatically builds the task model associated with a Web application, whereas Vaquita [15] and its evolutions build the concrete description associated with a Web page. In order to support the automatic redesign for migration purposes, we need to reconstruct concrete, abstract and task description. We have developed new transformations able to take Web pages and then provide any of the three possible logical descriptions.

At implementation level, the information regarding the concrete description is added to the abstract description. In fact, the concrete description is a refinement that adds information regarding concrete attributes to the structure provided by the abstract description. The abstract interface level represents platform-independent semantics of the user interface, thus the interface elements are identified in terms of the main effect that they support (for example: selection, editing, activator, navigator). It also provides a high-level description of how interactors (user interface elements) are arranged and composed together. The concrete interface level provides platform-dependent descriptions of the user interface and is responsible for how interactors and composition operators are implemented in the chosen platform with their related information content. For example, a selection interactor at the abstract level can be refined into a list or a radio-button (depending on the number of elements) in the case of a concrete description for a graphical desktop interface. The concrete description is still independent of the implementation languages. Thus, for example, the list could be still implemented in XHTML or Java or some other language for graphical applications.

The abstract description is used in the redesign phase in order to drive the changes in the choice of some interaction object implementations and their features and rearrange their distribution into the redesigned pages in such a way that the original semantics is still supported but using techniques suitable for the current platform. For example, a graphical list can be mapped onto an abstract selection element, which is then mapped onto a vocal enumerate choice in the case of migration from graphical desktop platform to vocal one. Both task and logical interface descriptions are used in order to create associations between the implementation elements in the original interface and in the redesigned one and restore the runtime state resulting from the user interactions with the source device in the corresponding elements of the target interface.

From the User Interface to its Logical Description.

The concrete interface is the logical level closest to the user interface implementation, it is organised in presentations interconnected by connection elements. Presentations are made up of logical descriptions of interaction objects called interactor elements. The interactors can be combined by composition operators. The goal of the composition operators is to identify the designers' communication goals, which determine how the interaction elements should be arranged in the presentation. We use four composition operators, which capture common effects that designers aim to achieve when they structure their presentations: Grouping (the elements are logically related to each other), Hierarchy (there are different levels of importance among elements), Relation (one element is related somehow to other elements), Ordering (when some order among elements can be identified).

We need to reverse the single desktop page that is accessed or that corresponds to the user interface part of a non-Desktop platform, which is currently accessed. The page is reversed into a concrete presentation and each interaction technique in the implementation into an interactor. The reversing algorithm works on the DOM tree of the page. In order to acquire it, we need to have well formed X/HTML files. Since many of the pages available on the Web do not satisfy this requirement, before starting the reversing phase, the page is parsed using the W3C Tidy parser that corrects features like missing and mismatching tags and returns the DOM tree of the corrected page. The corrected DOM tree is analysed recursively starting with the body element and going in depth. Each node is analysed by a specific function extracting information to generate the corresponding interactor or composition operator (Table 1).

Considering the X/HTML DOM node analysed by the recursive function, we have three basic cases:

- The node is mapped into a concrete interactor. This is a recursion endpoint. The appropriate interactor element is built and inserted into the XML-based logical description. For example DOM nodes corresponding to the tags <img>, <a> and <select> respectively cause the generation of a only_output – image, interaction - navigator and – selection interactor elements.
- The node corresponds to a composition operator. The proper composition element is built and the function is called recursively on the node subtrees. The subtree analysis can return both interactor and interactor composition elements. Whichever they are, they are appended to the composition element from which the recursive analysis started. For example the node corresponding to the tag <form> is reversed into a Relation composition operator, <ul> into Ordering.
- The node does not require the creation of an instance of interaction in the concrete specification (for example, in the Web page there is the definition of a new font). In this case no new element is added in the concrete description and if the node has no children we have a recursion endpoint. This can happen for example when there are some empty paragraphs <p></p> or line separators like <br> tags.

(X)HTML Element	Logic Interface Element
Unordered List	Ordering
Ordered List	
Div	Grouping
Fieldset	
Table	Description
Input checkbox	Single select
Input radio	Multiple select
Select	Single / multiple select
Form	Relation
Input text	Textfield
Input button	Navigator
Anchor	
Input reset	Activator
Input submit	Activator
Text	Text
Image	Object (Image)
Text+Image	Description
Text+Anchors	InteractiveDescription

Table 1. Transformation of Desktop Web implementation into Logical Description

From the User Interface Logical Description to the Task Model

A logical presentation can contain both elements that are single interactor objects and composition operator elements. The compositions can contain both simple interactors and further composition operators. Our reverse engineering transformation builds a task model represented through the ConcurTaskTrees (CTT) notation [16]. In this notation, tasks (which are the logical activities that should be accomplished in order to reach users' goals) are structured hierarchically, starting with high-level, general tasks, which are decomposed into more refined ones. A set of temporal operators are available to indicate the temporal relations among such tasks (sequential, concurrency, choice, disabling, suspend, order independence, …). A logical presentation is first associated with a high-level abstract task, which is decomposed into the subtrees obtained by reversing the elements contained in the presentation along with the appropriate temporal operators.

Each composition operator is reversed into a task, whose sub-tasks are the tasks obtained by reversing the elements to which the composition operator applies. Such subtasks are connected through appropriate CTT temporal operators. Each interactor is reversed into the corresponding basic task (a task that cannot be logically decomposed into subtasks). For example, a Relation composition in the user interface logical description corresponding to a Web form is reversed in such a way that all the basic tasks corresponding to elements involved in the Relation are connected by a concurrent temporal operator. Such elements are also associated with the task corresponding to the action triggering (for sending the form) by a disabling temporal operator.

Interface Redesign

The redesign transformation aims to change the design of a user interface. In particular, we propose a redesign for identifying solutions suitable for a different platform, which is performed automatically by exploiting semantic information contained in the logical description of the user interface (created by reverse process). Given the limited resources in screen size of mobile devices (such as cell phones or PDAs), desktop presentations generally must be split into a number of different presentations for the mobile devices. The logical description provides us with some semantic information that can be useful for identifying meaningful ways to split the desktop presentations along with the user interface state information (the actual implemented elements, such as labels, images, etc.). The redesign module analyses the input from the logical descriptions and generates an abstract and concrete description for the mobile device from which it is possible to automatically obtain the corresponding user interfaces. The redesign module also decides how abstract interactors and composition operators should be implemented in the target mobile platform.

Since our environment has to work on a single page at a time, it considers only its logical descriptions. It would be expensive in time and computational efficiency downloading all the site pages and reverse engineering all of them, especially considering sites composed of hundreds of pages.

Redesign for Small devices

In order to automatically redesign a desktop presentation for a mobile presentation we need to consider semantic information and the limits of the available resources. If we only consider the physical limitations we may end up dividing large pages into small pages which are not meaningful. To avoid this, we also consider the composition operators indicated in the logical descriptions. To this end, our algorithm tries to maintain interactors that are composed through some operator at the conceptual level in the same page, thus preserving the communication goals of the designer. However, this is not always possible because of the limitations of the platform. In this case, the algorithm aims to equally distribute the interactors into presentations of the mobile device.

In addition, the splitting of the pages requires a change in the navigation structure with the need for additional navigator interactors that allow access to the newly created pages. More specifically, the transformation follows the subsequent main criteria:

- The presentation split from desktop to mobile takes into account the composition operators because they indicate semantic relations among the elements that should be preserved in the resulting mobile interface.

- Another aspect considered is the number and cost of interactors. The cost is related to the interaction resources consumed, so it depends on pixels required, size of the fonts and other similar aspects.
- The implementation of the logical interactors may change according to the interaction resources available in the target platform (for example an input desktop text area, could be transformed into an input mobile text edit or also removed, because writing of long text is not a proper activities for a mobile device).
- The connections of the resulting interface should include the original ones and add those derived from the presentation split.
- The images should be resized according to the screen size of the target devices keeping the same aspect ratio. In some cases they may not be rendered at all because the result is too small or the mobile device does not support them.
- Text and labels can be transformed as well because they may be too long for the mobile devices. In converting labels we use tables able to identify shorter synonyms.

In particular, regarding the creation of new connections the following rules are applied:

- original connections of desktop presentations are associated to the mobile presentations that contain the interactor triggering the associated transition. The destination for each of these connections is the first mobile presentation obtained by splitting the original desktop destination presentation;
- composition operators that are allocated to a new mobile presentation are substituted in the original presentation by a link to the new presentation containing the first interactor associated with the composition operators.
- when a set of interactors composed through a specific operator has been split into multiple presentations because they do not fit into a single mobile presentation, then we need to introduce new connections to navigate through the new mobile presentations.

In the transformation process we take into account semantic aspects and the cost in terms of interaction resources of the elements considered. We have defined for each mobile device class identified (large, medium or small) a maximum acceptable overall cost in terms of the interaction resources utilizable in a single presentation.

Thus, each interactor and (even each composition operator) has a different cost in terms of interaction resources. The algorithm inserts interactors into a mobile presentation until the sum of individual interactor and composition operator costs reaches the maximum global cost supported. Examples of elements that determine the cost of interactors are the font size (in pixels) and number of characters in a text, image size (in pixels), if present. One example of the costs associated with composition operators is the minimum additional space (in pixels) needed to contain all its interactors in a readable layout. This additional value depends on the way the composition operator is implemented (for example, if a grouping is implemented with a fieldset or with bullets). Another example is the minimum and maximum interspace (in pixels) between the composed interactors.

After such considerations, it is easy to understand that each mobile presentation could contain a varying number of interactors depending of their interaction resources consumption. Table 2 shows how each desktop logical element is transformed for mobile devices.

Table 2. Transformation of logical interface for Mobile platform

Only output interactors	**Transformation for Mobile platform**
Text	Text is presented using a smaller font. In case the text is too long to be contained in a single page, the text is divided into more pages.
Description	The image is reduced in size while the text part is treated as normal text.
Object (image)	The image is reduced in size.
Interaction interactors	**Transformation for Mobile platform**
Navigator	Textual links are presented as are (changing fonts) or reduced in length, when they are too long, through the use of the synonym database. Image links have the image reduced in size.
Activator	Activator is transformed adapting the script code to the script supported by the mobile device web browser.
Text Edit	Text Edit is implemented through a *textfield* and if the specified length is too high, the length is reduced to fit the mobile device screen.
Numerical Edit	Same as Text Edit
Single selection	In dependence of the number of options, the implementing object can be a *radio button* or *select* element. In case the options are too many to be shown in a single PDA page, the selection menu is divided into sub menus or can be substituted with a textedit when the choices are intuitive.
Multiple selection	In dependence of the number of options, it can be implemented using a checkbox or select element. The number of choices should not be higher than the lines that can be shown in a single mobile device page.
Interactive Description	A piece of text containing links is generated applying separately the rules for text and link parts.
Composition operators	**Transformation for Mobile platform**
Grouping	Row grouping are transformed into column grouping and they are implemented through the use of fieldset element. The fieldset size is minimized in order to frame exactly the elements it contains.
Ordering	An ordered list is used.
Hierarchy	It is used a decreasing font and image size. The sizes are kept smaller than in the desktop version.
Relation	It is implemented through a form element.

Redesign for Vocal interaction

There are various differences to consider between graphical and vocal interfaces. In vocal interfaces there are several specific features that are important in order to support effective interaction with the user. For example, it is important that the system always provides feedback when it correctly interprets a vocal input and it is also useful to provide meaningful error feedback in the event of poor recognition of the user's vocal input. At any time, users should be able to interrupt the system with vocal keywords (for example "menu") to access other vocal sections/presentations or to activate particular features (such as the system reading a long text). An important aspect to consider is that sometimes users do not have an overall control of the system state, such as in graphic interfaces. In fact, short term memory can be easily disturbed by any kind of distraction. Thus, a useful technique is to provide some indication about the interface state in the application after a period of silence (timeout). Other useful support for this problem can be the use of speech titles and welcome or location sentences in each vocal presentation to allow users to understand their position and the subject of the current presentation and what input the system needs at that point. Another important difference between speech and graphic interfaces is that the vocal platform supports only sequential presentations and interactions while the graphical ones allow concurrent interactions. Thus, in vocal interfaces we have to find the right balance between the logical information structure and the length of presentations. For example, a large desktop presentation with some meaningful parts (i.e.: some top level grouping), it should not be transformed into a single vocal presentation containing a long sequential and heterogeneous vocal output. It is better to split desktop presentations by inserting elements associated with grouping composition operators in different vocal presentations so that the main vocal presentation is structured to access the various logical parts. On the other hand, newly created vocal presentations containing grouping content should not be further split in order to avoid a navigation level too deep, which could be a negative factor for user orientation in speech interfaces. Moreover, the generated vocal presentations representing the elements of each desktop grouping should produce the proper amount of homogeneous vocal flow. The main criteria of the redesign algorithm for the vocal platform are:

- Before redesign for vocal interaction, elements regarding tasks unsuitable for the vocal platform (for example, long text inputs) are removed and labels too long are modified (with the help of a database of terms suitable for vocal activities);
- Semantic relations among interactors in the original platform are maintained in the vocal platform, keeping interactors composed through the same composition operators in the same vocal presentation, and transforming them for better adaptation to the vocal device (see Table 1 below);

- Before redesign, images are removed, they are substituted by alternative descriptions (ALT tag). In some cases such descriptions can be obtained through OCR algorithms to recognize words and letters from the image content;
- Implementation of interactors and composition operators will change according to new vocal devices resources (see Table 3);
- The algorithm aims to provide a logical structure to vocal presentations avoiding too deep navigation levels because they may disorient users. To this end, the highest level composition operators (in the case of nested operators) are used to split desktop presentations into vocal presentations.
- Composition operators that are allocated to new vocal presentations are substituted in the main vocal presentation that cannot contain them by a vocal link to the new presentation, which contains the first interactor associated with the composition operator.
- Original connections of desktop presentations are associated to vocal presentations containing the corresponding triggering interactor; the destination presentation for each of these connections is the first vocal presentation associated with the redesign of the desktop destination presentation.

Table 3. Transformation of logical interface for vocal platform

Only output interactors	**Transformation for vocal platform**
Text	Text is read by system; generally long text reading, should be activated by the users to avoid long vocal feedback, which could annoy them. When possible too long text can be summarized or divided in structured paragraphs depending on their homogeneous content.
Description	Image transformed in text if present (or using content of Alt tag).
Object	Algorithm considers only alternative text.
Interaction interactors	**Transformation for vocal platform**
Navigator	*"Please, say X to access to…"* (where X represents the key word to activate the link)
Activator	*"Please, say X to activate the function …"* (where X represents the function to activate)
Text Edit	*"Please insert your " + label_value* (where *label_value* is the text label of the element contained in the desktop presentation; if the system understands from labels and synonyms database that label_value is a personal data like Name, Surname, Address, etc.), otherwise system asks: *"Please insert the " + label_value* (if the system understands that *label_value* represent a general data)
Numerical Edit	*"Please insert the " + label_value* (Similar to Text Edit, but input represents a number)
Single selection	The system read the question sentence of desktop selection and then the list of possible choices: X1, X2, X3; after that, system asks: *"Please make your choice."* (Number of choices should be a max number of four/five, because an high number of choices couldn't be memorized by user)
Multiple selection	It is transformed in a sequence of selections
Interactive description	Can be considered such as a text, but links should be managed in different

	ways: • if the text does not contain many links, keywords of each link can be hold between a specific sound or can be read with a loud tone; • if the text contains many links, all possible links can be listed at the end of text.
Composition operators	**Transformation for vocal platform**
Grouping	System inserts at beginning and at the end: a pause, a sound, a keyword or a change of volume
Ordering	System arranges elements in alphabetical order or with keywords
Hierarchy	System decreases or increase volume after each choice
Relation	System changes context

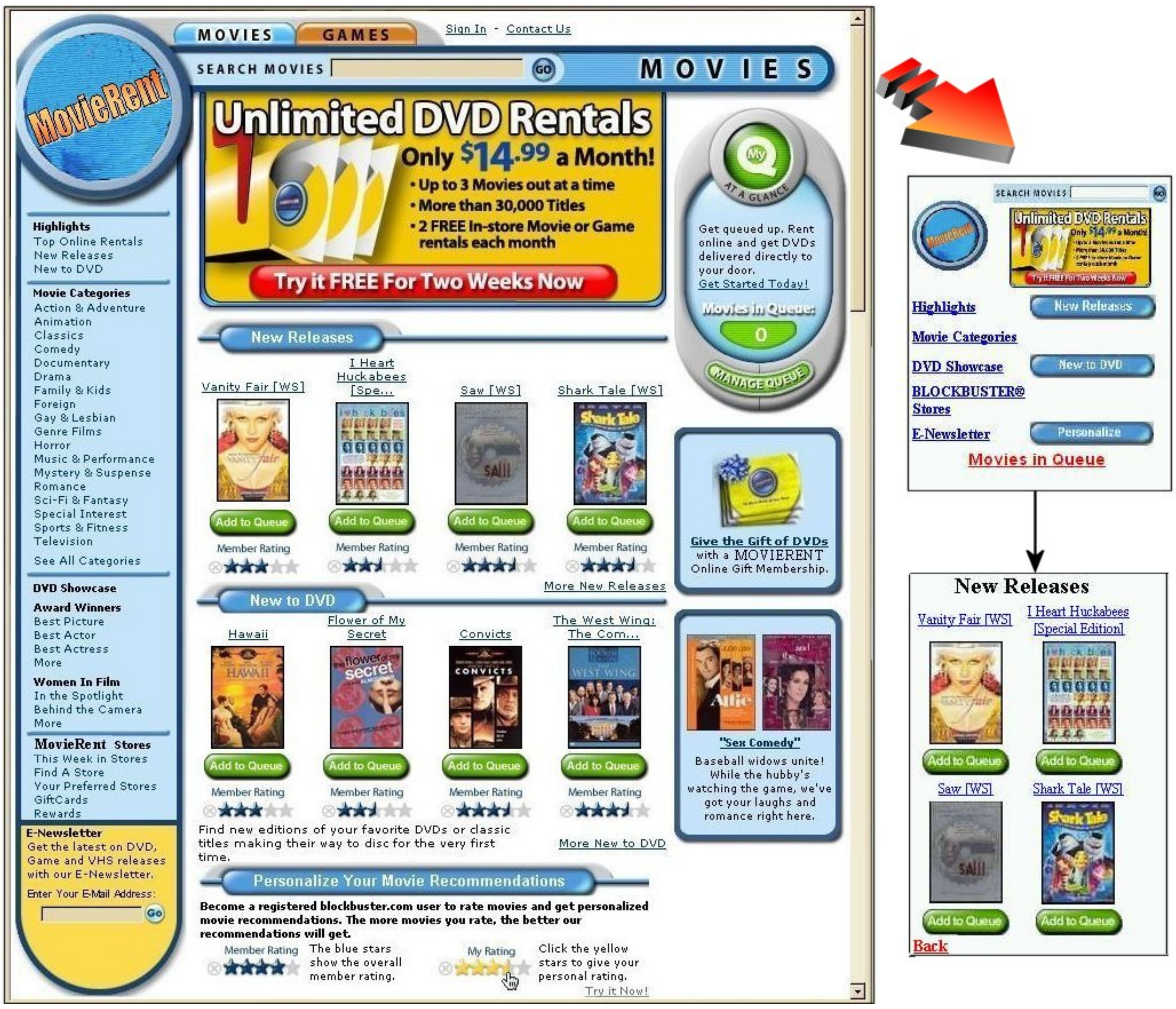
Engineered Support in Interface Redesign and Generation

In order to make our environment suitable for realistic contexts, with user sessions that require access to several pages of an application or users who access the same application at different times through different non-desktop platforms, sometimes even migrating through them, we had to consider various issues. One is related to the links of the generated pages for the target platform. It can happen that some of them are newly generated links while others are still those included in the original desktop version. Such links are modified in such a way as to redirect them to the proxy server so that, if selected, the corresponding page is loaded and transformed by the migration sever for generating new pages adapted to the target platform.

In order to improve the performance of our system, when logical descriptions of an accessed page are created then they are saved in the server so that if that page is accessed again through a non-desktop platform then the corresponding logical descriptions are immediately available and do not need to be calculated again. However, we have to take into account whether the considered page has been modified in the meantime because in this case the previously calculated logical descriptions can be no longer valid. Thus, the migration server compares the time of the last modification for the accessed page obtained from the content server with the time when the logical description was calculated, which is stored locally. If the page has been modified after its last reverse engineering then the corresponding logical descriptions have to be calculated again.

Sample Application

In order to explain how our environment works we introduce an example application. We consider a scenario and discuss the example, and how it is supported. The *MovieRent* site has been designed for a desktop platform. In our scenario the user accesses the site using a PDA. The migration/proxy server intercepts the PDA page requests to the *MovieRent* site, retrieves the Web pages and applies the reverse-redesign process to generate an interface version for the PDA platform. Before reverse engineering, the Proxy Server filters (when possible) some dynamic elements of the page (such as advertising, animations, old archive material, etc) and removes the elements implementing tasks not suitable for the new target platform. This last operation is performed with the help of an analysis of the type of interactor and the labels used (which can provide useful information regarding the associated task).



Get queued up. Rent online and get DVDs delivered directly to your door. Get Started Today!

Give the Gift of DVDs with a MOVIERENT Online Gift Membership.

Baseball widows unite! While the hubby's watching the game, we've got your laughs and romance right here.

Get the latest on DVD, Game and VHS releases with our E-Newsletter.

Find new editions of your favorite DVDs or classic titles making their way to disc for the very first time.

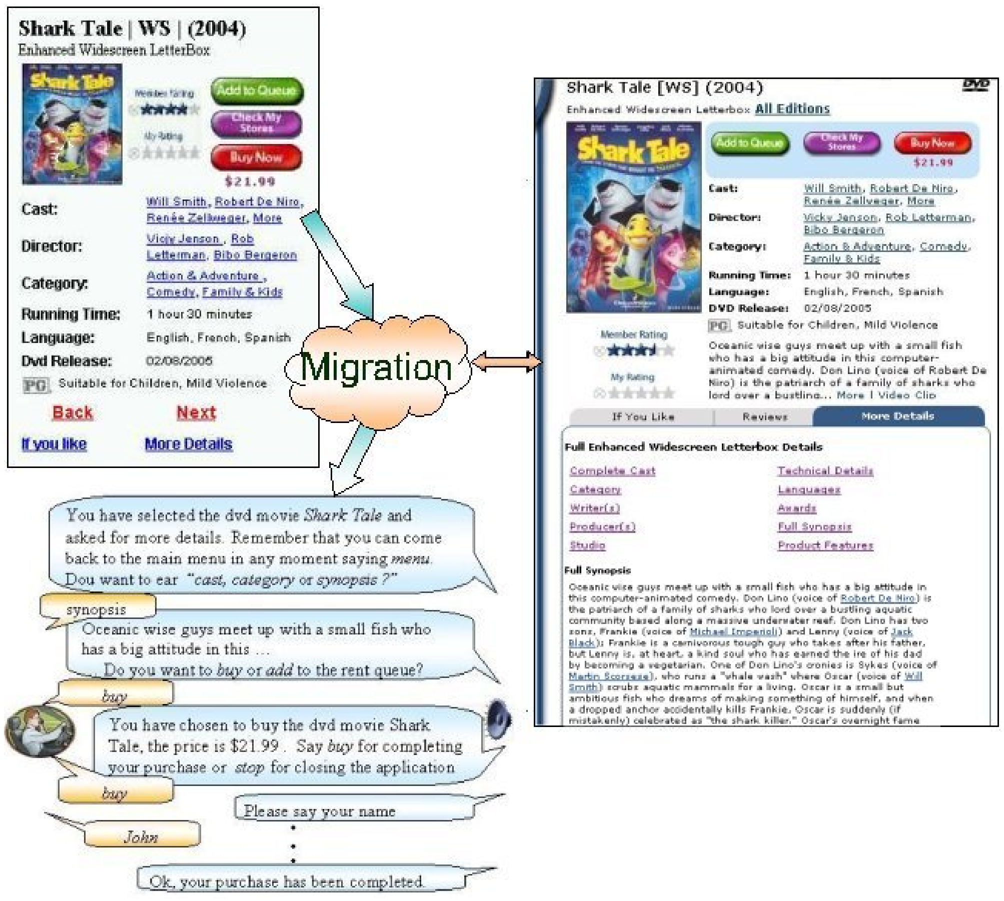
Become a registered blockbuster.com user to rate movies and get personalized movie recommendations. The more movies you rate, the better our recommendations will get.

The blue stars show the overall member rating.

Click the yellow stars to give your personal rating.

Figure 3. Desktop Page in the Example.

When the user selects the *Shark Tale* movie from the *New Release* page in the redesigned PDA page, the link still refers to the original page in the desktop version. The migration/proxy server retrieves it and produces the PDA interface shown in Figure 3. Notice that tabbed panels in the desktop version (Figure 4, right side) have been transformed into links in the PDA presentations (Figure 4, left side) and the *Reviews* tabbed panel has been removed because it contains elements for an activity, which is not particularly suitable for PDAs (long text insertion). Moreover, the left column containing links (Figure 3 left side) is repeated in all desktop pages, but to gain screen space they are inserted only in the main PDA presentation. Figure 4 also shows the central part of the Shark Tale DVD page for desktop. At this point of the scenario, the user enters his car and the application migrates to the voice car system. The migration/proxy server refers again to the original desktop page for building a vocal interface. The vocal interaction is represented in Figure 4 (on the bottom) where we can see the dialog corresponding to the desktop *Shark Tale* page, with the tab panel *More Details* selected and followed by the dialogue concerning the shipment and payment information and corresponding to the form that in the desktop version is loaded after clicking on the *Buy Now* button. When the user selects More Details on the PDA a new redesigned page is loaded. When migrating towards the voice system, the graphic to vocal redesign is done starting from the abstract description of the original page designed for the desktop platform, while the URL of the page loaded on the PDA is interpreted as interface state information indicating which tabbed panel must be visualized. When redesigning for the vocal platform, the tool generates a feedback message to remind the user which point of the interaction has been reached. The vocal interface continues to provide the movie information and enables the possibility of interrupting the active dialogue to access the menu dialogue through particular keywords (such as "menu"). Notice that the presentation of movie information in the vocal platform should not be too detailed. This is obtained by considering only important elements and filtering minor ones (such as running time, DVD Release, etc.), which are considered minor because of the limited space allocated in the graphical version. Thus, the vocal system provides the possibility of making a choice between *cast*, *category* and *synopsis*.



Shark Tale | WS | (2004)
Enhanced Widescreen LetterBox

$21.99

Cast:	Will Smith, Robert De Niro, Renée Zellweger, More
Director:	Vicky Jenson, Rob Letterman, Bibo Bergeron
Category:	Action & Adventure, Comedy, Family & Kids
Running Time:	1 hour 30 minutes
Language:	English, French, Spanish
Dvd Release:	02/08/2005

PG Suitable for Children, Mild Violence

Back Next
If you like More Details

You have selected the dvd movie *Shark Tale* and asked for more details. Remember that you can come back to the main menu in any moment saying *menu*. Dou want to ear *"cast, category or synopsis ?"*

synopsis

Oceanic wise guys meet up with a small fish who has a big attitude in this …
Do you want to *buy* or *add* to the rent queue?

buy

You have chosen to buy the dvd movie Shark Tale, the price is $21.99 . Say *buy* for completing your purchase or *stop* for closing the application

buy

Please say your name

John

Ok, your purchase has been completed.

Shark Tale [WS] (2004)
Enhanced Widescreen Letterbox All Editions

$21.99

Cast:	Will Smith, Robert De Niro, Renée Zellweger, More
Director:	Vicky Jenson, Rob Letterman, Bibo Bergeron
Category:	Action & Adventure, Comedy, Family & Kids
Running Time:	1 hour 30 minutes
Language:	English, French, Spanish
DVD Release:	02/08/2005

PG Suitable for Children, Mild Violence

Oceanic wise guys meet up with a small fish who has a big attitude in this computer-animated comedy. Don Lino (voice of Robert De Niro) is the patriarch of a family of sharks who lord over a bustling... More I Video Clip

If You Like | Reviews | More Details

Full Enhanced Widescreen Letterbox Details

Complete Cast	Technical Details
Category	Languages
Writer(s)	Awards
Producer(s)	Full Synopsis
Studio	Product Features

Full Synopsis

Oceanic wise guys meet up with a small fish who has a big attitude in this computer-animated comedy. Don Lino (voice of Robert De Niro) is the patriarch of a family of sharks who lord over a bustling aquatic community based along a massive underwater reef. Don Lino has two sons, Frankie (voice of Michael Imperioli) and Lenny (voice of Jack Black): Frankie is a carnivorous tough guy who takes after his father, but Lenny is, at heart, a kind soul who has earned the ire of his dad by becoming a vegetarian. One of Don Lino's cronies is Sykes (voice of Martin Scorsese), who runs a "whale wash" where Oscar (voice of Will Smith) scrubs aquatic mammals for a living. Oscar is a small but ambitious fish who dreams of making something of himself, and when a dropped anchor accidentally kills Frankie, Oscar is suddenly (if mistakenly) celebrated as "the shark killer." Oscar's overnight fame

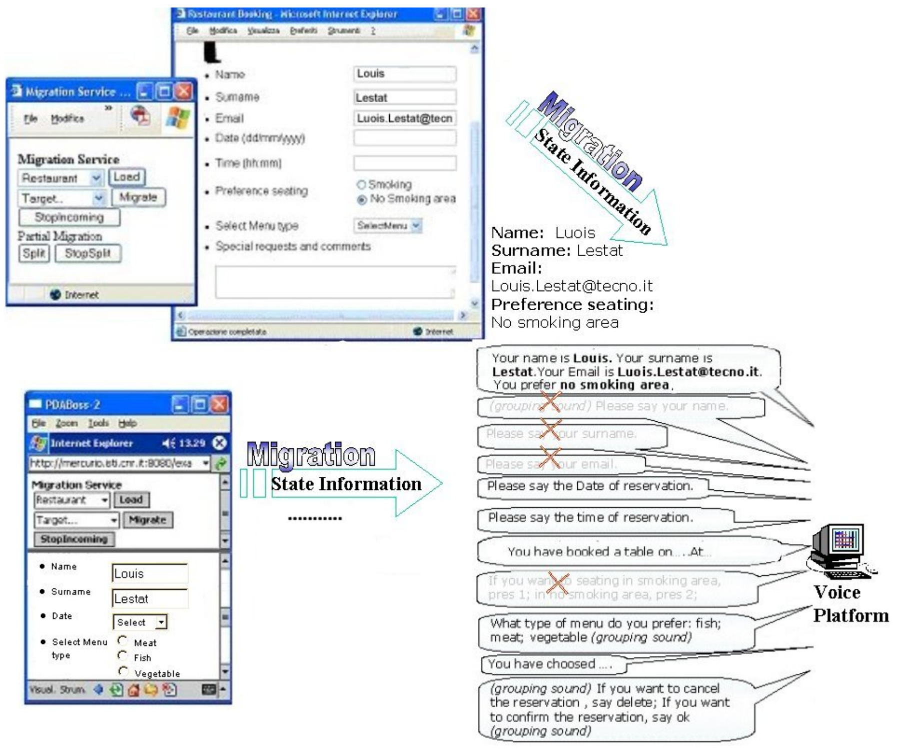
Figure 4. The Example Migration

User Tests

In order to understand the impact of migration on users and the usability of the interfaces obtained through the automatic redesign transformation some user tests were performed. The trans-modal migration funtionalities, from graphic to vocal, have been tested in the first version of the migration service.

Trans-modal migration test

The first test was performed on the "Restaurant" application, which allows users to select a restaurant, accessing its general information and make a reservation. The interfaces of the test application for desktop, PDA and vocal platforms differ both in the number of tasks and their implementation. For example, the date insertion is a text field in the desktop version and a selection object in the PDA version, while the insertion of free comments was removed from the vocal interface.



- Name: Louis
- Surname: Lestat
- Email: Luois.Lestat@tecn
- Date (dd/mm/yyyy)
- Time (hh:mm)
- Preference seating: Smoking [✓ No Smoking area]
- Select Menu type: SelectMenu
- Special requests and comments

Name: Luois
Surname: Lestat
Email:
Louis.Lestat@tecno.it
Preference seating:
No smoking area

- Name: Louis
- Surname: Lestat
- Date: Select
- Select Menu type: Meat / Fish / Vegetable

Your name is **Louis.** Your surname is **Lestat**.Your Email is **Luois.Lestat@tecno.it**. You prefer **no smoking area**.

~~*(grouping sound)* Please say your name.~~

~~Please say your surname.~~

~~Please say your email.~~

Please say the Date of reservation.

Please say the time of reservation.

You have booked a table on.....At...

~~If you want to seating in smoking area, pres 1; in no smoking area, pres 2;~~

What type of menu do you prefer: fish; meat; vegetable *(grouping sound)*

You have choosed .... .

(grouping sound) If you want to cancel the reservation , say delete; If you want to confirm the reservation, say ok *(grouping sound)*

Figure 5. Restaurant application and migration client interface

Since we were interested in considering multi-device environments, both a desktop PC and a PDA were used as graphic source platforms. This is useful for understanding if the features of the platform can influence the user because of the different interaction resources and, consequently, the different set of tasks supported. The 20 users involved were divided into two groups. The first one started with migration from PDA to vocal platform and repeated the experiment starting with the desktop. The second started with the desktop and repeated the test using the PDA. Users were asked to load the "Restaurant" application on the graphic device and start booking a table at a restaurant. At some point, they had to ask for migration towards the available vocal device and there complete the Restaurant Reservation task. After the session the users filled in the evaluation questionnaire. The average user age was 33.5 years (min 23 - max 68). Thirty percent of them were females, 65% had undergraduate degrees or higher and 55% had previously used a PDA. Users had good experience with graphic interfaces but far less with vocal ones: on a scale of 1 to 5, the average self-rating of graphic interface skill was 4.30 and 2.05 for vocal interfaces. For each migration experiment, users were asked to rate from 1 to 5 the parameters shown in Table 4.

Table 4. User rating for trans-modal migration attributes.

Parameters	Desktop to vocal	PDA to vocal
Migration client interface clearness	3.4	3.9
Interaction continuity easiness	4.35	4.65
Initial vocal feedback usefulness	4.1	4.2
Vocal feedback usefulness	4.25	4.25

Vocal feedback was provided via both the initial message, recalling the information inserted before migration, and a final message at the end of the session about the information inserted after migration. We chose this solution as the most likely to reduce user memory load. After the test, we asked the users if they would have preferred only total final feedback instead. Finally, we asked whether they noticed any difference between the graphic and vocal interface with the aim of finding out whether they could perceive the different number of supported tasks. The numeric test results were interpreted taking into account the answer justifications and free comments left in the questionnaire and considering user comments while performing the test.

Table 5. User preferences and salience of task differences.

Parameters	Laptop to vocal	PDA to vocal
Only final vocal feedback preferred	Yes 20% - No 80%	Yes 20% - No 80%
Noticed different task set	Yes 25% - No 75%	Yes 20% - No 80%

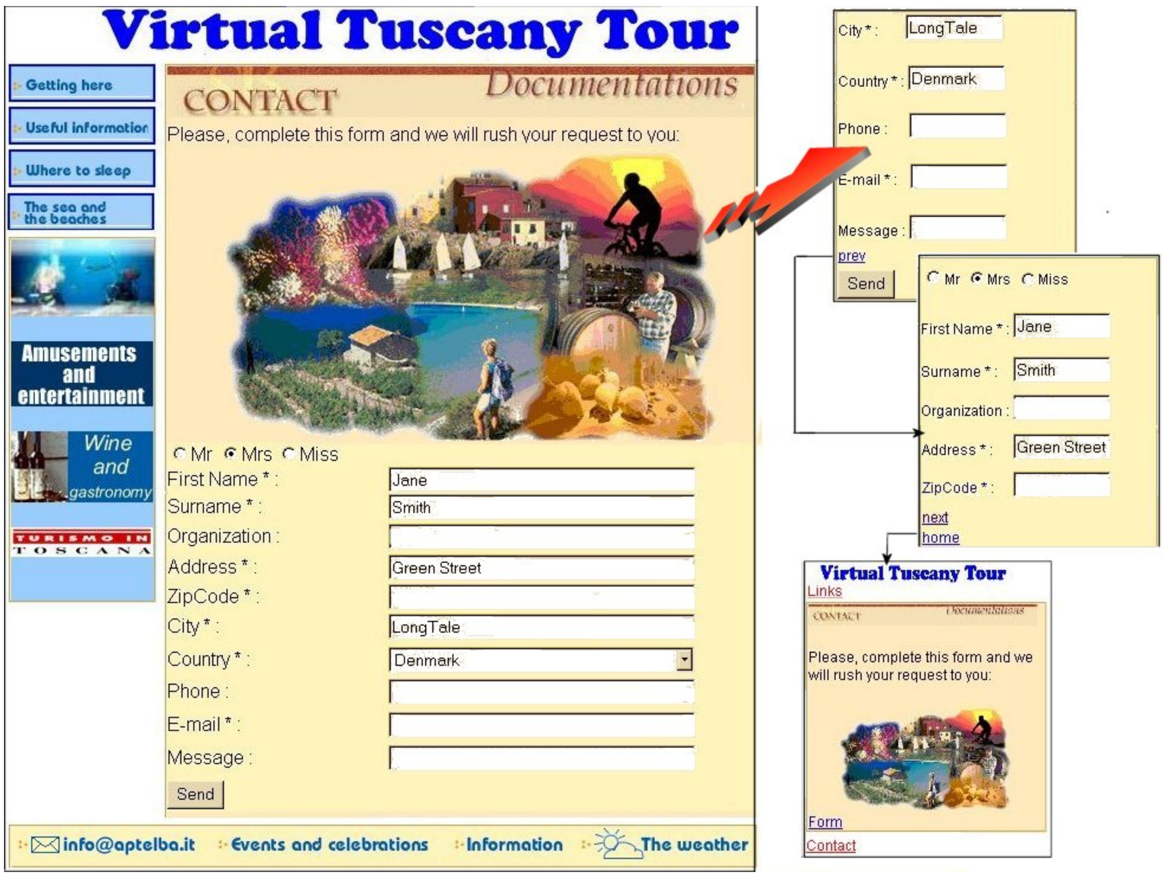
The service in itself was appreciated by users. Many judged it interesting and stimulating. The users had never tried any migration service before and interacted with it more easily in the second experiment, thus, showing it was easy to learn through practise, once the concepts underlying migration were understood. Interaction continuity received a slightly higher score in the PDA-to-vocal case. Indeed, the PDA and the vocal versions were more similar in terms of number of tasks than the desktop and the vocal ones. The difference in ease of continuity between the two platforms is small, thus the interaction continuity ease is influenced, but not compromised. Both the initial and the overall feedback through the vocal application were judged positively

(Table 4). The vocal feedback design was appreciated and 80% of the users would have not changed its style. One concern was the potential user disorientation in continuing interaction, not only by the change in modality, but also in the different range of possible actions to perform. Only 20-25% noticed the difference and it was perceived more in the desktop-to-vocal case (Table 5).

This first study provided some useful suggestions to keep in mind while designing user interface transmodal migration. The modality change does not cause disorientation but must be well supported by proper user feedback balancing completeness while avoiding boredom. The differences in interaction objects used to support the same task were not noticed at all, while the difference in the number of task supported was. Changing the number of actions that the user can perform can not be avoided due to the different capabilities of the platforms involved. However, this must be well designed in order to reduce as much as possible any sudden disruption in user's expectation.

Test on the new version of the migration environment.

The first test confirmed our choices concerning the support for trans-modal migration and gave useful suggestions for improving the migration environment. The work undertaken after the first test resulted in the new version of the migration environment discussed in this article and we performed a new test with a different application: the "Virtual Tuscany Tour". The goal was to have further empirical feedback on the migration concept and the new solution supporting it. Since very few changes concerned the trans-modal interface trasformation while many more affected the unimodal (graphic) migration, the new test considered desktop to PDA migration, in order to evaluate the new functionalities. The considered application is the Web site of a tourism agency specialized in trips around Tuscany, an Italian region. Interested users can ask material and detailed information about trips around the sea and the beach or around the countryside and mountains; it is also possible to have an overview of good quality places where to sleep. From this Web site it is also possible to get information about the main social events, amusements, places where to taste good food and wine, sport activities and other useful information, such as the weather forecast and public transportation. Thus, users during the test could freely interact with the application and at a certain point they were asked to fill in a form for requiring tourist documentation about Tuscany. The form had to be filled in partially, choosing some fields in any preferred order (see for example Figure 6). Hence users had to interact with the desktop migration client interface to require migration towards the PDA and continue the registration on the new device. They also were asked to check previously inserted information and performing some more tasks on the PDA. Lastly, users were asked to fill in an evaluation questionnaire.



CONTACT

Please, complete this form and we will rush your request to you:

Mr / [✓ Mrs] / Miss	
First Name * :	Jane
Surname * :	Smith
Organization :	
Address * :	Green Street
ZipCode * :	
City * :	LongTale
Country * :	Denmark
Phone :	
E-mail * :	
Message :	

Figure 6. The Test Application

The experiment involved 20 users whose average age was 31.94 years (min 22 - max 58). Forty percent of them were females, 85% were graduated or had a higher degree. Users had good experience in accessing Web applications through desktop platform and far less through PDA: on a scale of 1 to 5, the average self-rating of Web access through desktop platform skill was 4.35, while PDA usage ability was 1.95.

The evaluation questionnaire was divided into two parts, one concerning migration as a whole service and one related to the result of the automatic page redesign from desktop to PDA platform. The migration related part asked for a 1 to 5 score to the parameters shown in Table 6 along with the average values collected.

Table 6. Migration Service Evaluation

Parameters	Average Score
Migration Service Usefulness	4.1
Migration Client Interface Clearness	4.1
Easiness in Continuing Task Performance After Migration	4.4
Easiness in Retrieving Information Inserted Before Migration	4.75

User were also asked to say if they would have changed anything in the way migration is issued and only the 55% of them answered no. We discuss how this gave useful suggestions for improving the migration client interface in the next section. In Table 7 we show the average scores collected for the redesign part and the parameters we asked to evaluate, still on a 1 to 5 score scale.

Table 7. Redesign Transformation Evaluation

Parameters	Average Score
Visual Impact of PDA Redesigned Pages	3.9
Functional Navigation Structure on PDA	4.1
Interaction Difficulties due to Redesign Result	1.6
Interaction Difficulties due to PDA device	1.75

A final question concerning the redesign part was about noticing changes in the redesigned pages compared to the original ones. The 40% said yes, and we discuss what this means in the next section.

The evaluation questionnaire was not only a mere collection of numeric values because users were asked to provided justifications for each answer and free comments were also allowed. This gave us the possibility to better understand numeric results. Looking at Table 6, we can see that all the average scores concerning migration evaluation were higher than 4, moreover as we can see in Figure 7 scores below 2 did not pass the 10%. Some of the people who did not find useful the migration client said that they simply do not use any PDA.

The migration client interface resulted to be quite clear since it was considered basic and easy to use, some uncertainty was found just because it was never seen before, as some user commented, at a second usage they would have known what to do. The values about task performance continuity and access to information inserted before migration can be considered very satisfactory.

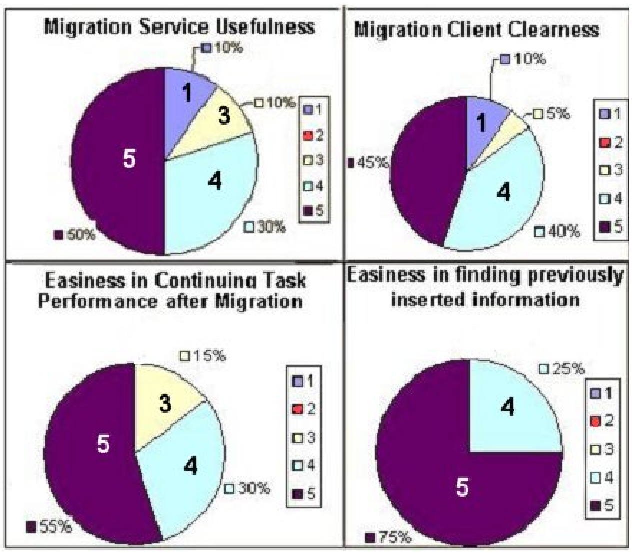
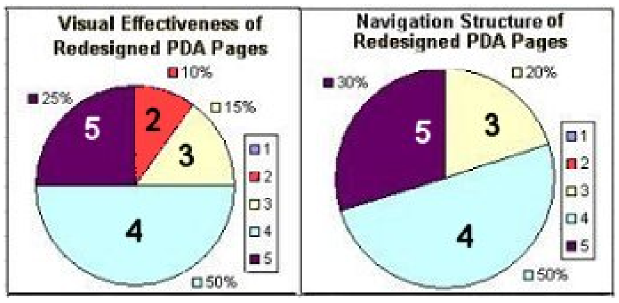
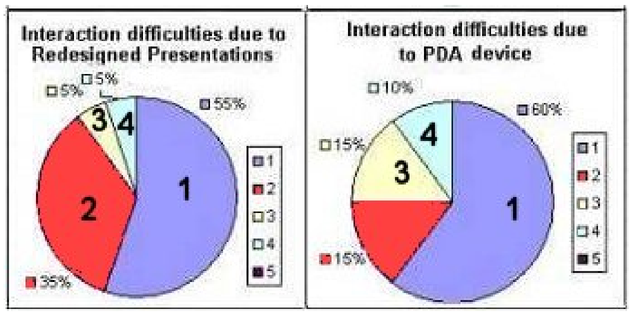


Figure 7. Migration Evaluation

Regarding the 45% of users who would have changed something in the way the migration was performed, this most referred to the list of IP addresses, which was used to indicate the possible target devices (instead of a significant label). We understand such criticism since the migration client interface was at prototypical stage and it is being improved in this respect.



(a)



(b)

Figure 8. Evaluation of the Redesign Transformation

Concerning the evaluation results obtained for the redesign part (see Table 7 and Figure 8a) we did not get the best results for the visual effectiveness of the redesigned pages. One issue was that current browsers for PDA have limitations with respect to those for desktop systems in terms of implementation constructs that are supported. The navigation structure of the redesigned pages was appreciated, we noticed that some people had difficulties in finding the links positioned at the bottom of the pages, while the links at the beginning of the page seemed easier to access. Indeed, some of the redesigned pages had links both on the top and on the bottom. Taking into consideration such difficulties we will consider the possibility of modifying the redesign transformation trying to obtain most links positioned at the beginning of the page.

About difficulties of interacting on the PDA platform (see Figure 8b), the numbers must be interpreted in a converse way, since the parameters are related to a negative feature. As we can see in Table 7, some difficulties were found while interacting with the PDA, but it must be said that most users had never used one before and many difficulties raised just because of low knowledge of this platform. Lastly, users said they noticed changes between the desktop and the PDA presentations. This question was made in order to understand if changing the specific interaction object implementing a given basic task (i.e.: a desktop pull down menu with hundreds of choices transformed in a PDA text input field) when changing platform could disturb the users. The comments provided in the questionnaire outlined that the changes that had been noticed were the obvious one regarding aspects related to the fact that a desktop single page had been split into multiple PDA pages and images where reduced in size. They did not make any observation concerning the different implementation of some tasks. Users did not find any particular difference that disturbed them.

Conclusions

We have presented an environment based on a migration/proxy server, which supports migratory interfaces in multi-device environments. Starting with a pre-existing version for desktop platform, the environment is able to dynamically generate interfaces for different platforms and modalities exploiting semantic information, which is derived through reverse engineering techniques. The environment currently supports access to Web applications and is able to generate versions for PDAs, various types of mobile phones and vocal devices and support migration through them. The application implementation languages currently supported in our GeReMi environment are XHTML, XHTML Mobile Profile and VoiceXML but we are extending it in order to also support X+V (for multimodal interaction), SVG (for direct manipulation graphical interfaces), Digital TV and gestural interaction with mobile devices. The tool has been tested on a number of Web applications (Movie rental, Tourism in Tuscany, Restaurant reservations, …). Our environment assumes that the desktop Web site has been implemented with standard W3C languages and filters some dynamic elements of the page (such as advertising, animations, …).

We also report on usability tests carried out in order to better understand the usability of migration and the interfaces resulting from our semantic redesign transformation. The results are encouraging and show that migration is a feature that can be appreciated by end users because it provides them with more flexibility in emerging multi-device environments, with a mix of stationary and mobile devices. The test results have also shown that the proposed redesign transformations are able to automatically generate presentations that allow users to be able to continue their task performance in the target device without being disoriented by the platform change.

Future work will be dedicated to identifying techniques for supporting distributing migration, where the user interface migrates from one device to multiple devices. This requires an intelligent processing able to identify usable solutions in splitting the resulting user interface across multiple devices at the end of the migration.

References

1. Puerta A., Eisenstein J., XIML: A Common Representation for Interaction Data. In *Proceedings of IUI2002: Sixth International Conference on Intelligent User Interfaces*, ACM, January 2002.
2. Abrams, M., Phanouriou, C., Batongbacal, A., Williams, S., Shuster, J., UIML: An Appliance-Independent XML User Interface Language, Proceedings of the 8th WWW conference, 1994..

3. Mori G., Paternò F., Santoro C. Design and Development of Multidevice User Interfaces through Multiple Logical Descriptions. *IEEE Transactions on Software Engineering* August 2004, Vol 30, No 8, IEEE Press, pp.507-520.
4. Limbourg, Q., Vanderdonckt, J., UsiXML: A User Interface Description Language Supporting Multiple Levels of Independence, in Matera, M., Comai, S. (Eds.), *Engineering Advanced Web Applications*, Rinton Press, Paramus, 2004.
5. Chen, Y., Ma, W.-Y., Zhang, H.-J. Detecting Web page structure for adaptive viewing on small form factor devices. In *Proceedings of the twelfth international conference on World Wide Web* **(***WWW'03)(*May 20-24, 2003, Budapest, Hungary), ACM 1-58113-680-3/03/0005, pp 225-233.
6. MacKay, B., Watters, C. R. Duffy, J. Web Page Transformation When Switching Devices. In *Proceedings of Sixth International Conference on Human Computer Interaction with Mobile Devices and Services (Mobile HCI'04)* (Glasgow, September 2004), LNCS 3160. Springer-Verlag, 228-239.
7. Nichols, J. Myers B. A., Higgins M., Hughes J., Harris T. K., Rosenfeld R., Pignol M. "Generating remote control interfaces for complex appliances". Proceedings ACM UIST'02, pp.161-170, 2002.
8. K. Gajos, D. Christianson, R. Hoffmann, T. Shaked, K. Henning, J. J. Long, and D. S. Weld. Fast and robust interface generation for ubiquitous applications. In Seventh International Conference on Ubiquitous Computing (UBICOMP'05), September 2005, pp. 37-55.
9. Ponnekanti, S. R. Lee, B. Fox, A. Hanrahan, P. and Winograd T.. ICrafter: A service framework for ubiquitous computing environments. In Proceedings of UBICOMP 2001. (Atlanta, Georgia, USA., 2001). LNCS 2201, ISBN:3-540-42614-0, Springer Verlag London UK. Pp 56-75.
10. Garlan, D., Siewiorek, D., Smailagic, A., Steenkiste, P. Project Aura: Toward Distraction-Free Pervasive Computing. IEEE Pervasive Computing, Vol 21, No 2 (April-June 2002), 22-31.
11. 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 the Second European Symposium on Ambient Intelligence (EUSAI* '04), LNCS 3295, Markopoulos et al. Springer-Verlag, Berlin Heidelberg, 2004, 291-302.
12. Bandelloni, R., Berti, S., Paternò, F. Mixed-Initiative, Trans-Modal Interface Migration. In *Proceedings of Sixth International Conference on Human Computer Interaction with Mobile Devices and Services (Mobile HCI'04)* (Glasgow, September 2004), LNCS 3160. Springer-Verlag, 216-227.
13. Bandelloni, R. Mori, G. Paternò, F., Dynamic Generation of Migratory Interfaces, Proceedings Mobile HCI 2005, ACM Press, Salzburg, September 2005.
14. Paganelli, L., Paternò, F. A Tool for Creating Design Models from Web Site Code. International Journal of Software Engineering and Knowledge Engineering, World Scientific Publishing 13(2), pp. 169-189 (2003).
15. Bouillon, L., Vanderdonckt, J. Retargeting Web Pages to other Computing Platforms. In *Proceedings of IEEE 9th Working Conference on Reverse Engineering* (WCRE'2002) (Richmond, Virginia, 29 October-1 November 2002), IEEE Computer Society Press, Los Alamitos, 2002, pp. 339-348.
16. Paternò, F. Model-based Design and Evaluation of Interactive Applications. Springer Verlag, ISBN 1-85233-155-0, 1999.