Push and Pull of Web User Interfaces in Multi-Device Environments

Giuseppe Ghiani, Fabio Paternò, Carmen Santoro
CNR-ISTI, HIIS
Via Moruzzi, 1
56124 Pisa, Italy
{giuseppe.ghiani, fabio.paterno, carmen.santoro}@isti.cnr.it

ABSTRACT
In this work we present 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.

We describe mechanisms for sharing information regarding devices, users, and Web applications with various levels of privacy and report on first experiences with the proposed environment.

Categories and Subject Descriptors
H.5 Information Interfaces And Presentation; H.5.2 User Interfaces.

General Terms
Design, Human Factors.

Keywords
Migratory Web User Interfaces, Multi-device Environments, Continuity.

1. INTRODUCTION
Recent years have witnessed the increasing availability in the mass market of various types of interactive devices with a wide range of interaction resources and the advent of social computing, which means that more and more people use their interactive applications not in isolation but together with friends and colleagues. Such trends have obviously involved Web applications, which are the most common. Thus, there is a need to support users in their shared interactions in the emerging ubiquitous environments. For this purpose, we have designed and developed a novel solution, which is able to support migration of interactive Web applications when multiple users are sharing multi-device environments. Such solution allows users to migrate applications (or parts of them) from one device to another and continue their tasks from the point they left off. This is becoming more and more important because of the increase of computing devices per person and the need for people to exploit such technological offer while freely moving, especially when accessing applications requiring long sessions (such as games, e-commerce, specialized services, …). In addition, it also allows users to pull applications from other devices in case they think they can be useful for them.

We support migration through an environment that poses no constraints in terms of authoring environments to use when developing Web applications that can exploit migration.

The goal of such environment is to provide flexible support in various aspects taking into account the variety of possible devices that users may want to employ. Thus, it also allows users to interactively select the parts of the user interface to migrate, which is a useful feature when complex applications are accessed and devices with limited screen size are used as migration targets. It requires a minimum effort by users: they just have to access a migration client (still a Web application) through their browser. This migration client allows them to log into the migration environment, and 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. Thus, migration can occur through personal and public devices. The major research question of this study is whether such migration platform is a usable and useful tool to better support seamless Web application access in multi-device environments.

In the paper, after discussing related work, we introduce a couple of example scenarios in both business and personal domains. Then, we provide a description of the environment for managing migration in multi-user and multi-device environments and the associated privacy policies, including some architectural details on its components and their relations. Then, we report on a user test done with this multi-user migration environment. Lastly, some conclusions along with indications for future work are drawn.

2. RELATED WORK
An initial framework for cross-platform service user experience was proposed in [12]. 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 matches the user's expectations); continuity, and consistency. The study described considers accessing contents and functionalities through various platforms at different times. The solution we present 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 [4], found

AVI '12, May 21-25, 2012, Capri Island, Italy

out that users employ a variety of techniques for accessing information across devices. However, there is still room for improvement, since participants reported managing information across their devices as the most challenging aspect of using multiple devices. Thus, migratory interfaces can be an important support from this viewpoint.

While Pick-and-Drop [11] 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 [7] put forward ISR (Internet Suspend Resume), a migration solution that encapsulates all volatile execution state of a virtual machine. However, the incompatibility of the virtual machine between different platforms leads to the major limitation of this approach: 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 [8]. Instead, we aim to allow migration of any Web application developed with standard Web languages, without using any further, specific add-on. Page Tailor [2] is a tool for reusable end-user customization for the mobile Web: since we enable end users to decide what components to migrate to mobile devices, we thus integrate customization and migration.

Quan et al. [10] 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, we support the possibility of pausing the execution of a task, and then afterwards being able to resume and continue it.

Some concepts that are exploited in our migration approach are also used in Highlight [9], a server-side architecture that enables rapid prototyping and deployment of mobile Web applications. With Highlight, existing Web sites (also including dynamic JavaScript and AJAX) are re-authored to support smaller, task-specific interactions through embedding a Web browser inside a proxy server and using a remote control metaphor where the mobile browser controls the proxy browser. In our approach the migration 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 [1] 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. In this area an interesting contribution was WebSplitter [6], 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 have to specify the Web content in XML and define a policy file indicating what content should be shown for each device and user. However, it does not allow preserving the state when people and applications change device. In the area of collaborative Web browsing, PlayByPlay [13] aims to support collaborations among users by means of Web page annotations. We do not address collaborative browsing but have instead focused on creating a general, browser-independent environment for supporting multi-user migration, without requiring additional software installation (such as plugins, etc.). Collaboration among users within Web applications was previously addressed in [5]. In that system users surf the Web through a proxy server, which –as in our case– adds some JavaScript code to the navigated pages in order to add a toolbar to control collaboration. In our solution, we preferred not modifying the original layout of the page (which may be confusing for users), then we chose to place the Migration client (used for accessing the migration environment) on a separate browser tab. Also, rather than supporting co-browsing (considered in [5]), we support state-persistent migration, 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 [1]. However, while [1] focuses on how the JavaScript state is preserved in mono-user Web migration, in this paper we present and discuss several new platform 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; and support for privacy/security.

A framework for task migration across devices, named Deep Shot, has been recently proposed [3]. 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 but in order to make an application migratory, some functionality of the Deep Shot framework must be manually integrated at developing time. In addition, for Web applications a specific browser-specific plug-in should be installed. We aim to provide a solution accessible through any browser-enhanced device able to preserve the application state on the client side (e.g., values of form fields, JavaScript variables, cookies).

3. EXAMPLE SCENARIOS

In this section two typical migration scenarios supported by our environment are provided. They deal with two main migration modalities: *push* and *pull*. The former is the forwarding of Web pages from the device the user is using 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 send them towards the current user's device. In both cases the state of the interactive application is preserved.

3.1 Scenario 1 (Professional Domain)

Barbara, Christian and Marco work for a technological company based in London, which is participating to an exhibition in Milan. Since they have to attend the event, they need to find a suitable flight and accommodation for the trip. They would like 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. Meanwhile, Barbara and Christian can look at his Web activity, since all of them are registered users on this 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 realised 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 accepts such a "pull" migration 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. Indeed, 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).

Nevertheless, 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 our migration platform which, differently from other solutions, provides state persistence and cross-device continuity for Web interactive applications.

3.2 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 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 (e.g. advertisement banners with dynamic background image).

So, 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 on her network only the most interesting ones. 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 towards his device an interesting article on that topic.

4. 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, …) can access the platform services in spite of having different operating systems, interaction resources, etc. In order to overcome this issue, our choice 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 and no explicit user intervention is needed to extract the state from the source page and restore it in the target one.

4.1 The Architecture

Figure 1 provides an overview of the architecture of our environment. First the users register the devices to the migration platform. The Migration Client is a Web application for managing some of the migration platform functionality. It includes access to a device discovery protocol (see arrow 1 in Figure 1), and lets the user launch any Web application (2) through the Migration Proxy (3, 4), which annotates them with JavaScripts that are 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. The upload is carried out in three steps (see Figure 1): first an incoming migration message is sent by the Migration Server to the Migration Client of the target device (arrow 7) 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 sections.

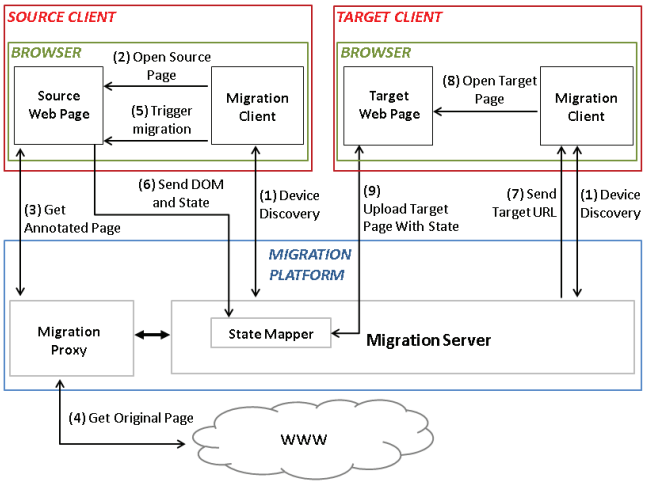

Figure 1. The Architecture of the Multi-Device Environment.

4.2 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 1), by including additional JavaScript code. Such modification is needed in order to make the accessed page suitable for the migration platform. Indeed, when accessing a Web page, 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 such added 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/) our platform is able to handle secure connections with Web pages accessed through the HTTPS protocol, by saving the digital certificate in the migration server.

4.3 Migration Client

As we mentioned before, 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 1) passing through the migration Proxy. According to user preferences, each page is launched by the Migration Client in a separate browser tab or 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 2) provides various pieces of information: the Web pages that the user is navigating, and 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.

When the user decides to trigger a migration, s/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 2).

Upon migration request, a tiny window pops up, enabling the user to choose the actual target device(s) among the available ones (see Figure 3). It is worth noting that the only available devices are the ones that belong to the user, the ones belonging to other users and have been set by their owners as publicly visible and targetable, and the public devices.

As previously mentioned, the component selection on a navigated page is enabled from the Migration Client when selecting partial migration. The Migration Client scripts are 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 access 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.

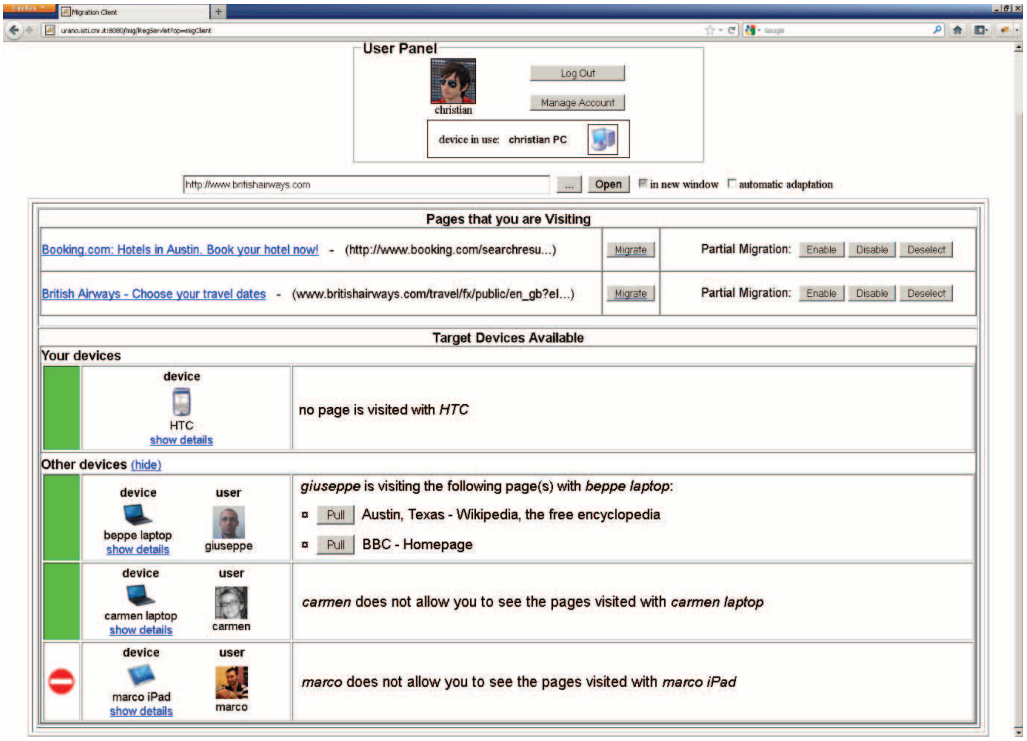

Figure 2. The Migration Client.

In this case it is not possible for the Migration Client to access the migratory (injected) functionalities of the navigated Web page, because, due to the different protocols, any access would cause a cross-domain exception. 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 case 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, …).

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 scripts added by the migration proxy are actually *concatenated* to the ones that manage the original events of the page. This is done in order to preserve the original behaviour of the page. Furthermore, after migrating parts of a page towards another device (partial migration), the user can continue the interaction from such migrated parts in the same way as it occurs after having performed a total migration.

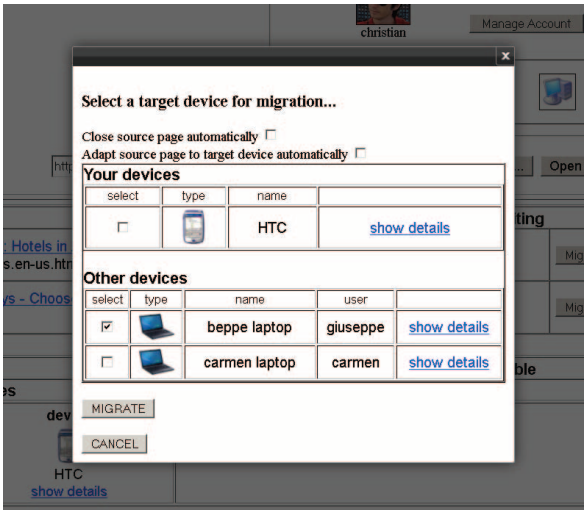

Figure 3. The Trigger Support in the Migration Client.

4.4 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. Through this page, devices can be added, deleted and configured with different levels of privacy. The configuration (see Figure 4) comprises several parameters whose values indicate the level of privacy of the related device. In particular, we have:

- *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.

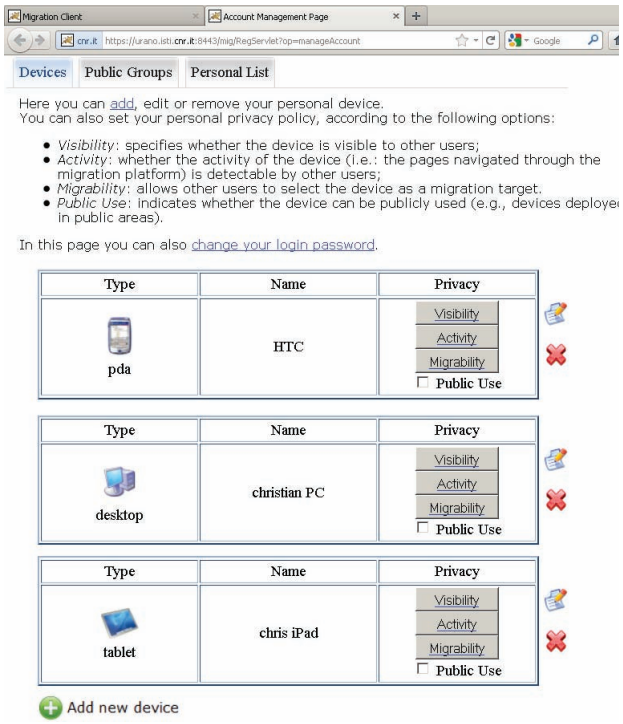

Figure 4: The Account Management page.

For each of the above mentioned parameters, several values are possible: *public* (everybody has the right), *private* (only the device owner has the right) or *limited* (only a subset of users has the right). 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 his/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, detectable and migrable for everyone belonging to the friends group.

4.5 DOM Capture and 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. Our 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 actually 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 is able to serialize 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. This 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.

5. EVALUATION

A user test was performed in our 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 we did not consider any related measurement.

5.1 Test Organisation

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 the participants. 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.

5.2 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 our research institute, but they were not members of our laboratory. Also, 5 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 our migration platform.

5.3 Scenario and Tasks Performed

In the proposed 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 described before, they are supposed to arrange a trip abroad, and then 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, in the scenario performed 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 the moderator. 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.

5.4 Results

For the test, 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.

5.4.1 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.

Usefulness of 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 (i.e.: the session was not maintained).

Usefulness of total push migration 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 it also includes 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 with interesting information.

Usefulness of 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.

Usefulness of privacy policies - Usefulness ([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 multi-user migration platform should be fully aware of the fact that pages can be pushed/pulled from/to her device.

5.4.2 Usability

Usability of 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 user 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).

Usability of 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.

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

Usability of 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 our 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.

Usability of 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 4 (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 suggested to integrate the settings panel with meaningful icons.

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

5.4.3 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 because they had explicitly requested that page from another device. We still think that it would be better to notify the user that a new window/tab is being opened, but we can make this feature optional in the next version 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 centring the confirmation box and using a blinking style for 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.

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.

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 our 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 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 appreciate the possibility to access the platform by any standard browser, without having to install any application or plugin.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a platform for managing migration in multi-user and multi-device environments, and the associated privacy policies, also reporting on a user test. In general, the environment is able to migrate the Web pages through both push and pull modality without posing any constraint on their authoring and requiring any manual modification. In a few cases there are some problems in preserving all the state, for instance when the pages use JQuery to dynamically modify the DOM.

The user test carried out gave us interesting and promising indications on the general reaction of users with respect to usability and usefulness of the proposed platform, and also enabled us to identify the parts that need further consideration, also with respect to more technical indicators (e.g. performance time).

We have already implemented some of the suggestions provided by the test participants (e.g. increase the visibility of the panel for accepting the migration), while we plan to address other suggestions (improve the account management page layout, make optional the migration notification, ...) into an improved version of the platform. Efficiency aspects will be tackled too, as we believe that scalability is relevant when large groups of users access the platform at the same time. Moreover, we also plan to perform further empirical validation with the new version.

7. ACKNOWLEDGEMENTS

We gratefully acknowledge support from the EU ARTEMIS SMARCOS Project (http://www.smarcos-project.eu).

8. REFERENCES

1. Bellucci, F., Ghiani, G., Paternò, F., and Santoro, C. Engineering JavaScript State Persistence of Web Applications Migrating across Multiple Devices, Proceedings EICS 2011, Pisa, June 2011, ACM Press.
2. Bila, N., Ronda, T., Mohomed, I., Truong, K. N., and de Lara E. PageTailor: reusable end-user customization for the mobile Web. MobiSys 2007: 16-29.
3. Chang, T.H., and Li, Y. Deep Shot: A Framework for Migrating Tasks Across Devices Using Mobile Phone Cameras. Proceedings ACM CHI '11, pp. 2163-2172.
4. Dearman, D., and Pierce J. "It's on my other Computer!": Computing with Multiple Devices. Proceedings ACM CHI '08, pp. 767-776 (2008).
5. Ding, Y., and Huber J. Designing multi-user multi-device systems: an architecture for multi-browsing applications, Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia, 2008.
6. Han R., Perret V., and Naghshineh M. WebSplitter: a unified XML framework for multi-device collaborative Web browsing. CSCW 2000: 221-230, ACM Press.
7. Kozuch, M., and Satyanarayanan M. Internet Suspend/Resume, Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'02) IEEE Press, 2002.
8. Melchior, J., Grolaux D., Vanderdonckt J., and Van Roy P. A toolkit for peer-to-peer distributed user interfaces: concepts, implementation, and applications. Proc. EICS 2009, pp. 69-78.
9. Nichols, J., Hua, Z. and Barton, J. Highlight: a System for Creating and Deploying Mobile Web applications. In Proceedings UIST'08. ACM, New York,, 249-258.
10. Quan, D., Huynh, D., Karger, D. R., and Miller, R. User interface Continuations. In Proceedings UIST '03. ACM, New York, 145-148.
11. Rekimoto, J. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. Proceedings of UIST'97, pp.31-39, (1997).
12. Wäljas, M., Segerståhl, K., Väänänen-Vainio-Mattila, K., and Oinas-Kukkonen, H. Cross-Platform Service User Experience: A Field Study and an Initial Framework. In Proc. of MobileHCI '10, ACM.
13. Wiltse, H., and Nichols J. PlayByPlay: collaborative Web browsing for desktop and mobile devices. CHI 2009: 1781-1790, ACM Press.