J Ambient Intell Human Comput
DOI 10.1007/s12652-013-0214-7

ORIGINAL RESEARCH

# Evaluating context-aware user interface migration in multi-device environments

**Giuseppe Ghiani · Jussi Polet · Ville Antila · Jani Mäntyjärvi**

Received: 15 June 2013 / Accepted: 25 November 2013


**Abstract** People access applications and services through different devices depending on location and needs. During a single day, people can use a smartphone, tablet, PC and a TV, sequentially or simultaneously, depending on the context of use. Even within a single task, people often start with one device, such as a smartphone, and continue with another device, such as a PC, as the task evolves. To provide fluent task continuation, the system should provide ways to migrate on-going sessions from one device to another in a seamless manner. In this paper, we present a solution for migrating user interface and maintaining the interaction session across devices when changing situations. With two studies we gain insights into user needs and technical requirements for context-aware information sharing in multi-device environments. A longitudinal diary study was conducted to uncover specific situations where users have needs for information sharing, and how they would prefer the system to react in those situations. We also conducted a controlled user study using a prototype system for session migration between devices in changing contexts, with three different operational modes: manual, assisted and automatic, to gain a deeper knowledge into the requirements. The findings indicate a need for easier interaction whilst switching between devices and that these needs are often situation-specific. We also report in detail how people would prefer the system to perform migrations automatically and intelligently suggest them in some situations. Moreover, we draw technical requirements for such a system in order to develop seamless context-aware migration.

**Keywords** Context-awareness · Device ensembles · Migration · Migratory user interfaces · Multi-device environments

## 1 Introduction

The usage of applications and services has been more and more fragmented over different kinds of interactive devices in different situations depending on the current location, social surroundings and needs. In addition, the ubiquity of computing brings users to diverse types of device ensembles at home, work as well as on the move. During a normal day, one can use a tablet computer when eating breakfast, a smartphone on the bus, a laptop computer at work and an interactive large public display when out in the city—to just read the news.

Also, reading news or browsing the *Web* is not the only task fragmented over different devices but also more complex and demanding tasks, such as email and working on documents. Now, the flow of interaction in each case, regardless of the task at hand, is always interrupted when changing between devices. Moreover, the data required by each application and service is often not in sync between these different devices, thus resulting in a time-consuming and tedious task of synchronisation.

*"Very often I need to share some information, Web pages or files, that are stored on a laptop or desktop*

**Electronic supplementary material** The online version of this article (doi:10.1007/s12652-013-0214-7) contains supplementary material, which is available to authorized users.

G. Ghiani (✉)
ISTI-CNR, Via G. Moruzzi 1, 56124 Pisa, Italy
e-mail: giuseppe.ghiani@isti.cnr.it

J. Polet · V. Antila · J. Mäntyjärvi
VTT Technical Research Centre of Finland, Kaitoväylä 1, 90571 Oulu, Finland

*with my mobile device but it takes too much time to complete the task or it is sometimes impossible. In particular, if I have to repeat the operation several times a day it is quite annoying."*—Participant #2, male, 26 years old.

Currently, the common solution for having your data synchronised between all your devices is to have the data in the cloud—documents, music, photos and videos etc. However, the remaining challenge is to offer a consistent transition from one device to another in terms of the *user interface* (UI) itself and the application session or state information. Another key challenge is to provide the user support in this task, to ease the cognitive load of the transition. Thus, easy and intelligent migration of the whole *user experience* (UX) is needed between devices.

To address these challenges, we have conducted two studies focusing on exploring the user needs and requirements for context-awareness in *multi-device environments* (MDE). The approach for this research is two-fold; we conducted a longitudinal *diary study* to investigate the needs for information sharing in MDEs focusing specifically on the needs for context-awareness. In addition, to obtain deeper knowledge on the users' attitudes towards the context-awareness, we conducted *user tests in a controlled environment*, comparing different ways of operating a prototype context-aware system.

Today's MDEs are characterised by the needs of task continuity and continuous movement, therefore the prototype presented in this paper is targeted specifically towards those. Our results indicate that there are important correlations between the technical qualities of the system and the perceived usability and UX. We also found out that the users are willing to allow the system to have context-aware, automatic behaviour in some circumstances and environments.

In the following sections, related work on multi-device interaction and interaction techniques will be discussed as well as the benefits of context-awareness in MDEs. After that we present the methodology for the user studies and analyse the key findings. Finally, discussion and conclusions are drawn from the study results and summary of the main contributions of this research are given.

## 2 Related work

A recent research (Google 2012) studied the multi-device usage and the related behaviour of the users, and identified two main interaction ways with device ensembles: *sequential* and *simultaneous*. The research results indicate that the prevalence of multi-device usage, whether it is sequential or simultaneous, makes it imperative that e.g. businesses should enable the customers to save their shopping carts and provide "signed-in" experiences or the ability to email the progress to them when changing between devices. The research results also indicate that the choice of the devices used at a particular time is very often driven by the context and situation. Therefore, it can be stated that there is a demand for intelligent, context-aware functionality for migrating applications and UX between devices, in addition to the overall demand for "intermodal UX" (IUX), i.e. UX that spans over multiple heterogeneous devices. The "cross-screen engagement" of consumers has also been studied (Microsoft 2013) and the results show that "multi-screening" is increasingly the default operating mode for consumers.

Another recent study (Forrester Research 2013) also investigated the different places vs. the use of different screen sizes (smartphone vs. tablet). The report implies that there is a clear transition in the devices used when people are on the move vs. at the office vs. at home as well as difference on a room-level too (kitchen vs. living room etc.). Similar to the conclusions given by Microsoft, it has been stated that the demand for "always-on services" is growing as they seek out for different devices for different kinds of needs (videos vs. email, at home vs. on-the-go).

In the past, multi-device interaction has also been investigated. Dearman and Pierce (2008) focused on techniques people use to access multiple devices and argue that one of the core problems is how to support seamless device changes. This is the main challenge that we are studying and aiming to tackle by providing an integrated, context-aware functionality for device changes in order to try to distil the user needs and requirements for such functionality. As Dearman and Pierce also suggest, there are opportunities to improve the IUX by focusing on the user rather than on applications and devices. Therefore, we aim to find the user needs and requirements for the functionality proposed in this study.

Migrating a UI between devices can be done either by "pull" migration, where the user initiates the migration at the target device, thus "pulling" the UI from the source device into the target device. Similarly, the migration can also be "push" migration, where the user "pushes" the UI from the source device into the target device. In this study, we do not explicitly distinguish the trigger originating device, as we want to focus on the context-awareness dimension. But in general, in the assisted and automatic modes the mobile device performed the migration triggering, either by pulling or pushing the UI depending of the use case scenarios, which are later explained in detail.

Dees (2011) implemented different prototypes and utilised them to conduct three user tests to study migratory (nomadic) UIs' usability. He found out that people like the pull type of migration more than the push type, because of the premature commitment that the push paradigm entails, whereas interactions between devices, in domestic environments, are more ad-hoc in nature. Also, the device availability and reachability issues were seen as a problem in the push migration. A need for users to have explicit control over UI suspension, rather than the system doing this automatically, was also revealed. He found out that the system should preserve as much state information as possible in the migration (e.g. page/UI position, text field inputs etc.). Ghiani et al. (2012b) also considered push vs. pull in migration, but focused on a business scenario and discovered that most users considered the push as the most useful default modality. A possible reason for this may lie in the type of task of the study, that involved multiple users (i.e. simulated job colleagues). Participants performed single target and multiple target migrations between their devices and devices belonging to others. Multi-user capability could have affected the way participants perceived security and privacy issues of the pull mode, and making them prefer the more explicit push mode. Overall, it seems that the preference of modality is use-case specific and related to the user being in control of the migration.

UI *consistency* in the migration was also studied by Dees and, similarly to Pyla et al. (2006), he found out that consistency is an important factor, but can sometimes be prioritised lower in favour of more important factors, such as activity *continuity*. More importantly, he was also able to classify some of the user needs (in domestic environments) based on the experiment; the main needs being the mobility (migration abilities, access from anywhere and suspension abilities) of the activities across devices as well as the possibility to enhance the action on some device with the capabilities of another device (e.g. UI from smartphone to a large screen for leveraged capabilities, appropriate and adapted style of interaction as well as the possibility for collaboration).

Overall, to ease the access to information through multiple devices, support for migratory UIs is needed. Wäljas et al. (2010) conducted a few weeks long user study and studied how various multi-device applications are accessed by people, and thus proposed an initial framework for analysing cross-platform service UX, which relates to the two main aspects already mentioned, continuity (fluency of the content and task migration), and consistency, as well as a third main aspect, *composition*. This also relates to Dees's notion of enhanced interaction; devices can have different roles and the functionality can be distributed in order to achieve the best possible UX with a composition of devices, e.g. a smartphone can retain the controls of a video player but the video itself could be migrated to a TV screen nearby.

As mentioned, Pyla et al. also discuss the seamless device change from the perspective of task continuity and state that seamless task continuity is a more important factor than the consistency of the UI across devices and platforms. They state that the disconnection point occurring in a task when changing from one device to another can be mitigated by utilising interaction techniques that leverage the unique capabilities of each platform, instead of trying to forcefully preserve the consistency on each device.

Fischer et al. (2009) have dealt with co-location of devices available in the environment. Their system, called *RELATE*, helps users in discovering devices nearby. For determining the relative position of available devices, the system relies on ad-hoc localisation hardware. We believe that such context-awareness can enhance nomadic users' multi-device experience. As discussed by Brdiczka (2010), context data can be used by means of evolving situation models to infer high level situations in which the users are involved and this idea matches with ours. However, rather than in designing/developing custom hardware, we are indeed more interested in how the system behaviour affects users' perceived usability.

Differently from RELATE, in this study we rely on the embedded capabilities of consumer devices. Widely available communication technologies that can be used for sensing co-location of target devices include *near field communication* of *radio frequency identification* and *Bluetooth* (Cheverst et al. 2005; Rekimoto et al. 2003). Techniques that exploit the proximity factors of computing devices, (e.g. location/position, orientation, movement) have been investigated, e.g. in ubiquitous computing environments (Greenberg et al. 2011). Such factors are indeed considered to facilitate user-device interaction, for instance by providing the system with knowledge about the context and letting it take decisions on input and/or output strategies (Ballendat et al. 2010; Marquardt et al. 2011).

From the technical viewpoint of embedded devices a smooth context-aware UI migration during an interaction session requires that some core technical functionalities such as context recognition, setup of communication, data transfer, application launches with interface rendering and outputs indicating successful migration should have minimum delay. E.g. Könönen and Pääkkönen (2011) have studied the prediction of communication timer alignment for enabling more energy efficient functionality of always-on applications. They discuss that by predicting behaviour and executing some communication functionality initialisations beforehand, the delays for accessing the service is much shorter. This kind of technical functionality is very important for achieving enjoyable IUX for context-aware UI migration.

To mention some of the recent technical and commercial developments in the field of IUX, include e.g. the Google *Chrome* Web browser's ability to save and synchronise the browser tabs between multiple devices[1] (e.g. the user's PC and mobile device browsers). Google Chrome extension named *Chrome to Mobile*[2] allows the

[1] https://support.google.com/chrome/answer/2591582?hl=en-GB.

[2] https://support.google.com/chrome/answer/2451559?hl=en-GB.

sending of Web pages from desktops to mobile devices. Compared to Chrome's tab synchronisation and Chrome to Mobile extension, our ultimate goal is to allow not only Web page/session migration from desktop to mobile devices between any connected devices, as well as to enhance the migration functionality with context/situation-awareness—to enable the device ensemble to be intelligently proactive. In general, with this study we aim to investigate the impact of such intelligent functionalities on how users perceive application and data migration related aspects. Google has also introduced a new version of their *Analytics*, now called *Universal Analytics*, which includes a *multi-platform tracking* capability to gather analytics data across multiple devices, which could enable gathering valuable information for overall IUX needs. Multiple companies have also started to incorporate and offer IUX with their products e.g. *MOVL Connect Platform*[3] for smartphone and TV multi-device usage and *Pocket*[4] (formerly known as *Read It Later*), which is kind of an evolved version of *Dropbox*, allowing the user to save any content for later across multiple devices. Whereas MOVL is targeted for smartphone and TV related multi-device usage, our goal is to include any capable devices into the system. Pocket on the other hand is focused on data sharing alone but we aim to enable the sharing and migration of both applications (and their front-ends) and data.

The issues of differences in UI rendering due to device diversity and possible adaptations of UI to a destination device (e.g. presentation components rearrangement) are not considered in this paper. Other more technical issues, such as security and privacy implications, when migrating interaction sessions across devices, were previously discussed by Ghiani et al. (2012a). Instead, the ability to use a simple sensing application as an enabler of context-awareness in a MDE is demonstrated and users' attitudes towards such automatic supports are studied. In detail, the proposed system is able to automatically detect some context variables, such as user location, device position, information about the user's current activity (e.g. walking or still) and combinations of these. The possibilities of using this context information to allow the system to perform automatic or semi-automatic (assisted) migration of session from one device to another are discussed. This type of automatic trigger can be related to the work on implicit *human–computer interaction* driven by the context discussed by Schmidt (2000), in which the system acts proactively on the basis of context information.

The research community is showing great interest in Context Aware Recommender Systems (CARS), i.e. in how to exploit information with some degree of certainty to trigger/suggest actions. Adomavicius and Tuzhilin (2011) state the relevance of several context dimensions, such as user location, that we also rely on. Our prototype for context-aware migration in some way pertains to CARS, as it is able to "recommend" migration in assisted modality. However, our work is more devoted to study users experience rather than to reason on context modelling. Doryab et al. (2012), though not explicitly mentioning CARS, consider action recommendation in a clinical environment and highlight the importance of user location/activity recognition in this regard. A major difference with our work is that, due to the criticality of hospital scenarios, the system does not act automatically based on the context, and its intervention is limited to suggesting virtual actions to the user.

## 3 Approach and methodology

The goal of this study was two-fold, (1) to gain information about the needs for intelligent, context-aware information sharing, (2) to understand the expectations and acceptance of system behaviour in different situations (e.g. home vs. office).

First of all, to understand the user needs for information sharing in MDEs, we conducted a longitudinal study ($N = 24$) where the participants logged their needs for sharing information between devices.

Secondly, the objective is to a gain deeper understanding of the requirements for context-aware information sharing in MDEs. We conducted a lab study ($N = 24$), in which the users performed a series of tasks related to session migration from one device to another using a prototype system. In this test setting the users migrated a Web application session from one device to another. The system had three operational modes: *automatic*, *assisted* and *manual* operation. Two of these modes were context-aware. In automatic mode, when a predefined trigger for migration is sensed by the system, it performs this action automatically. On the other hand, in assisted mode, when a trigger for migration is sensed, the system notifies the user of this possibility, leaving the final decision to migrate to the user. The goal of this study was to compare the different operational modes of this system to gain an understanding of the expectations and requirements for context-awareness in MDEs. In the following sections we describe the specific methodology followed in both phases of the research approach.

### 3.1 Methodology

In as heterogeneous trial group as possible, a total of 24 persons, were recruited, consisting of both computer science professionals and of people who only use computers and smartphones for basic office work purposes. The trial

[3] http://connect.movl.com/.

[4] http://getpocket.com/.

participants were recruited by sending an email within our organisations, asking for interested people, who matched our recruitment criteria, to participate into our study. The same people participated in both study phases: the *laboratory* and the *diary* study phase.

### 3.1.1 Test setup and phases

First, we organised the laboratory user tests. The user tests were scheduled for a time span of 1.5 weeks, approximately three to four participants per day, maximum 60 min each. The test tasks consisted of Web page migration from a smartphone to a PC and vice versa using three different interaction modes (manual, assisted and automatic). When diverse modes are available to operate an interactive system, "mode errors" may occur. As discussed by Sellen et al. (1992), these are situations in which users forget the mode in use. In our test setting the system did not provide any explicit mode-dependent feedback. However, we did not expect to have mode errors because the users were provided with instructions for each task and informed about the current mode before starting each interaction.

In addition to a background questionnaire, we had the participants fill an in-between questionnaire after each migration mode (manual, assisted and automatic).

The laboratory studies were also recorded on video to count the focus shifts between the devices and to catch any freeform comments and suggestions during the laboratory tests. By focus shifts, we loosely mean the *macro attention shifts* described by Holleis et al. (2007).

After the laboratory user tests, we instructed the participants for the *diary study phase*. In the diary study, the trial participants took notes of their everyday situations, in which they would find context-aware information sharing between devices to be useful and to specify between which devices. The notion of "device" in this context was loosely-defined to include also imaginary situations where such information sharing would not be technically possible with the current technology.

The diary entries were done by filling in a Web form. On the Web form the participants were asked to fill-in the following characteristics of the situation:

- Source device
- Target device
- Type of information that has been exchanged between the devices
- The on-going task: what were you doing at the time? What were you trying to achieve?
- Place where the situation happened
- Description of the situation
- If these devices would have a "shared memory", how would you like the information to be exchanged between the devices?
- What did you do to cope with the situation (or what do you currently do in these kinds of situations)?

The diary study phase was approximately 2 weeks long for each participant, the actual length spanned from 2 weeks to 2 months between participants.

### 3.1.2 Demography and background information

The 24 test participants were on average 32 years old (median 31 years), the youngest participant being 26 years old and the oldest participant 44 years old. Only four female participants were involved in the study. Most of the people were Finnish (11) and Italian (11) by nationality, but also one German and one Portuguese participated. Education level was high: 6 bachelors, 13 masters and 5 doctors. 2 participants were partially impaired (one in wheelchair and one without a hand). Approximately half of the participants were "tech-savvy" (developers, early adopters and highly interested in technology) and the rest had "normal technical skills" (i.e. basic knowledge in information technology).

As shown in Fig. 1 left, 12 participants used two to four computing devices (computers, smartphones, smart TVs, tablets etc.) regularly (on a daily basis at home and work etc.), 9 participants used 4–6 computing devices regularly, 2 participants used over 6 computing devices regularly and only one participant used 1–2 computing devices regularly. Figure 1 right shows the mediums used for sharing information between devices; 20 participants used email as a sharing medium between their devices, 21 used some online file transfer service (cloud services such as DropBox, *File Transfer Protocol* clients etc.), all of the participants used some physical medium as a sharing solution (external hard drive, *Universal Serial Bus* stick etc.) and 11 participants used some online collaboration tool for sharing information (*Evernote*, Google *Docs* etc.). Additionally, Bluetooth was regularly mentioned as a sharing tool between devices as well as *Internet Relay Chat*, *Microsoft Exchange Active Sync*, some social media applications and even *UNIX* shells with screens.

## 3.2 Prototype system

The prototype environment used in the laboratory study is an extension of a previously developed multi-device Web-based *Migration Platform* (Ghiani et al. 2012b). In previous studies with the prototype system, it was noticed that

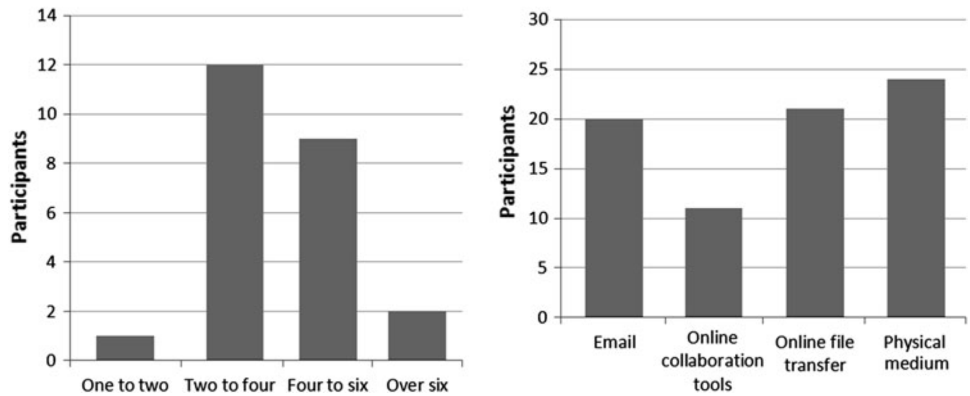



**Fig. 1** Number of computing devices used regularly (*left*) and usual tools for information sharing between devices (*right*)

the perceived usability was highly related to the selection efficiency of the target device (target acquisition) and the process initiation (triggering). By making the system context-aware, these two aspects can be enhanced by the system offering relevant choices for target devices and by suggesting or initiating the process automatically, as discussed by Schilit et al. (1994). The selection of relevant target devices and initiation of the migration process is enabled by the system being aware of the surrounding devices, environment, users and social relations. Thus, the main benefits of making a multi-device system context-aware are:

- *Automatic target acquisition and selection*—with the system knowing and using information of the environment, the migration target device can be chosen or recommended;
- *Suggested or automatic migration*—with the system knowing the current situation, it can either suggest the migration to a nearby device (assisted migration) or trigger the migration automatically (automatic migration).

### *3.2.1 General architecture*

The prototype system is based on four main components; (1) devices, (2) *Migration client* accessed through the device Web browser, (3) *Context monitor*[5] running on the smartphone and (4) Migration Platform. The Migration Platform acts as a proxy server, which annotates existing Web pages in real-time, when a user is browsing the Web (accessed through the Migration Client).

An overview of the system architecture is shown in Fig. 2, where the environment, involving two devices, and main connections among the modules are depicted.

Figure 3 shows a sequence diagram of the communications occurring between the architecture modules in migration. Devices A (smartphone) and B (desktop computer) have access to Web through the Migration Platform proxy. Both devices are running the Migration Client that manages authentication on the Platform (1, 2). The smartphone is running the Context Monitor on the background. A Web page is opened through the Migration Client (i.e. by providing its URL in the navigation form) on Device A, the request goes (3) through the proxy server. The request is released (4) to the actual application server, which then responds back with the resource to the proxy server. The migration proxy annotates the received resource [e.g. a *hypertext mark-up language* (HTML) page] by adding additional *JavaScript* code enabling the migration and sends the annotated resource as a response to Device A (5) The user interacts with the page and, if needed, browses the links. It is worth noting that, even if a link is activated, the navigation still goes on through the Migration Platform proxy (thus again through steps 3, 4 and 5). This is because all the links within the annotated page are enhanced with the proxy base URL.

In this example, the Context Monitor of Device A is aware of the prevailing situation. Depending on the parameters in use and its configuration, e.g. by knowing the motion of Device A, its proximity to a potential target device, and the preferences of the user, the system can sense opportune moments for migration (e.g. Device A is in the close proximity of Device B and placed display-up on a flat surface, such as a table). When an opportune moment for migration is sensed, e.g. when some relevant context event is detected (6), the Context Monitor of Device A informs the Migration Platform (7), which notifies the Migration Client of the source device (8) by requesting the current page content. The source Migration Client processes the content to be migrated to the target device by invoking the previously injected JavaScript excerpt on the browsed page document. The excerpt serialises the page *document object model* (DOM), including the interaction state. This data is serialised and sent to the Migration Platform (9), which saves them locally. The reason why three steps (7, 8 and 9) are needed instead of one (7) for the Platform to obtain the current page to migrate lies in the dynamicity of the page itself: Web pages

[5] https://github.com/cavtt.

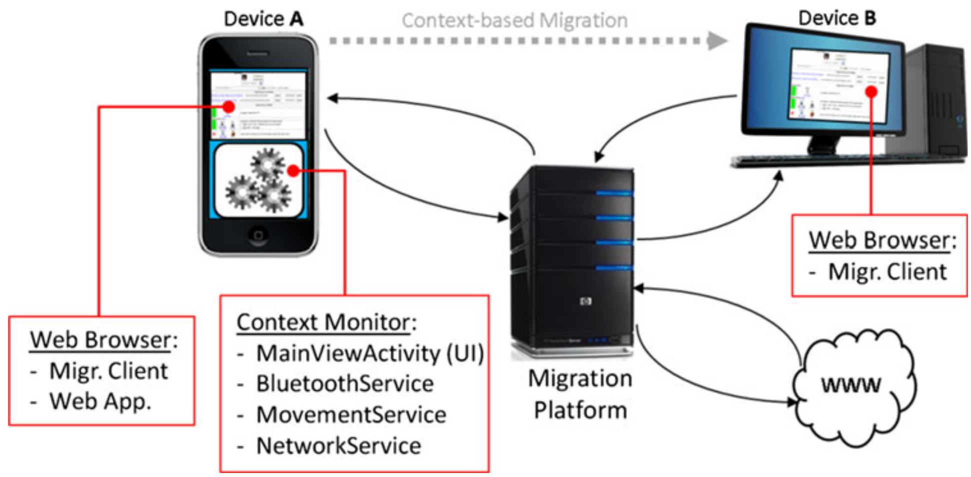



**Fig. 2** Overview of the system architecture

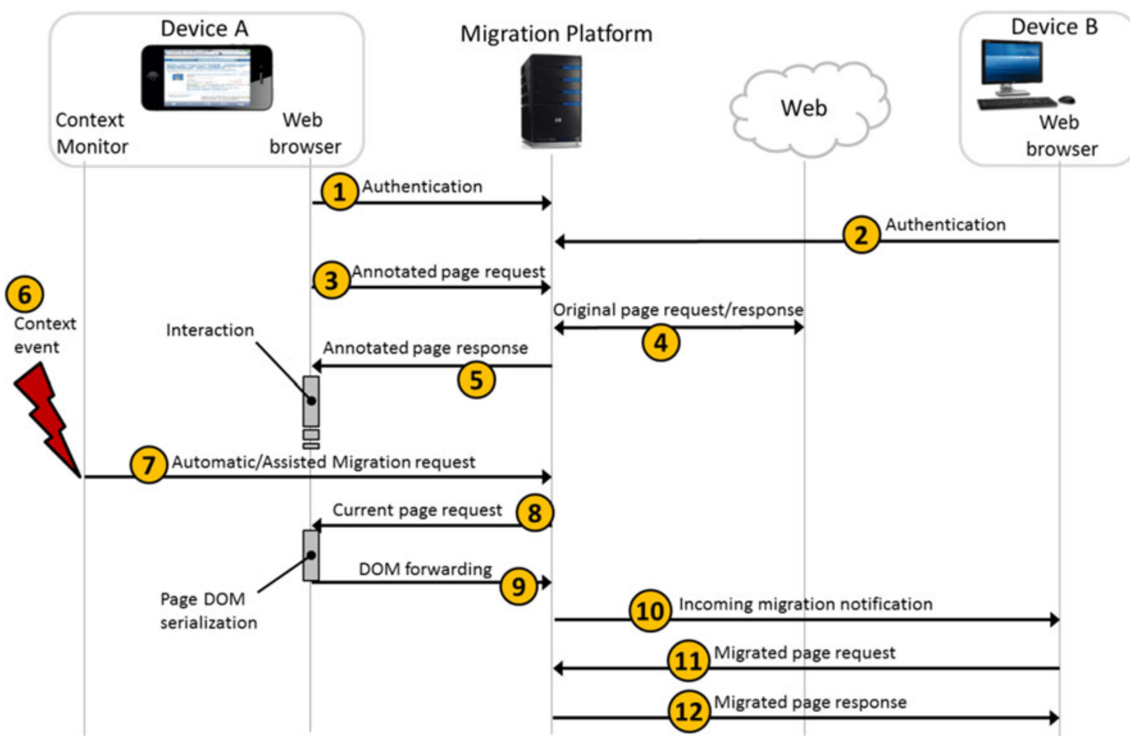



**Fig. 3** Sequence diagram of performing a migration

are often interactive and their content can change depending on user interaction and/or dynamic content loading (e.g. advertisements). Instead of the page cached in the Platform proxy, the DOM in the originating device browser should be used as starting content for migration. This is because the former simply defines the page HTML code downloaded from the application server, while the latter embeds the result of all user interactions.

The Platform then notifies the Migration Client running on the target device (10) and provides it with the address to access the Web page to be migrated. The Migration Client of Device B automatically opens the migrated page on the Web browser. A full architecture description of the Migration Platform is presented in Ghiani et al. (2012b), where the issues of page annotation, state persistence and preservation of the original functionalities are tackled in depth. In general, most issues were related to the need of having cross-platform and cross-browser compatibility. Another requirement was that the Web applications had to be potentially migratory without the need of being modified server-side. All the enhancements needed to be done, on-the-fly, in the Migration Platform proxy. Due to the number and complexity of such technological problems, we are not discussing them again here. The same applies to the security and privacy concerns of the Platform, discussed in Ghiani et al. (2012a). Most of them were related to the usage of an intermediate proxy both during Web navigation and in page migration, ranging between confidentiality of data exchanged between devices through the proxy and management of session persistence in multi-user migration.

### *3.2.2 Context-awareness*

The context-awareness of the prototype system is enabled via the so-called Context Monitor, a software running on the smartphone. The Context Monitor is a mobile application developed for *Android* smartphones, which runs on the background and observes the selected context parameters. The context parameters can be e.g. the proximity of other devices [obtained by using the Bluetooth *received signal strength indicator* (RSSI)], the current physical activity of the user (fall detection, still, walking and running activities obtained by using the device accelerometers), the current virtual activity of the user [the currently running applications and currently used applications on the device, call and *short message service* activities etc.], location [*global positioning system* coordinates, *wireless local area networks*, *base station cell ID*] and other, such as audio and light etc.

The context parameters can be chosen and configured, so that the user can set when the Web application should be migrated from the smartphone to another device and vice versa. Thus, e.g. the user can set the target devices for the migration and also the context-dependent rules, such as "perform migration from the smartphone to the office computer when the smartphone is still after being placed on a table, facing upwards and in the close proximity of the office computer" and "perform migration from the office computer to the smartphone when the smartphone has been lifted from the table, is moving, and the office computer is not in close proximity".

It needs to be noted that the Context Monitor is a dedicated mobile application developed for the sole purposes of this study. More specifically we wanted to investigate the user acceptance of different technical, performance-related metrics, such as latency, speed, accuracy and their possible correlation to user evaluations. The main focus of the study was to compare three different operational modes of the system: *manual*, *assisted* and *automatic*. Therefore, the functionality of the Context Monitor was fine-tuned to fit the test scenarios present in the user tests to enable fair comparison between the different operational modes. The application does not aim to tackle the problems related to the wide field of activity recognition, as those questions are out of the scope of this paper.

### *3.2.3 Test scenarios*

There were two test scenarios that the users performed in the laboratory study. Both of these scenarios were performed using all three different operational modes. To minimise any learning effects, we shuffled the migration modes for each test participant. By permuting the manual, assisted and automatic modes, we obtained six different orders in which the users performed the following scenarios. For the context-awareness of the system we focused on the scenario-specific factors such as the smartphone stability: detecting if the phone is placed on the table and the presence/proximity information: detecting whether the phone is in a certain environment. The two test scenarios are the following.

**Fig. 4** Test scenario overview: phone placed on the table near the PC

*3.2.3.1 Scenario 1* The user is approaching the office while browsing some Web page on their Android smartphone's Web browser, accessed through the Migration Client. The Context Monitor software is running on the phone, detecting the presence/proximity (Bluetooth) of their work desktop computer as well as the stability and orientation of the phone. As soon as s/he gets to their desk and put the smartphone on the table (see Fig. 4), the browsed Web page is migrated, depending on the test mode, manually, automatically or in assisted mode, from the phone browser to the computer browser.

Since migration is always triggered from the source device (i.e. the smartphone) towards the target one, this is a typical pushed migration scenario.

*3.2.3.2 Scenario 2* The user is surfing some Web application (accessed via the Migration Client) on the desktop computer and suddenly needs to leave his office for a meeting. S/he takes the smartphone with them and walks away to move to the meeting room. The Context Monitor notices that the smartphone is now on the move and not facing upwards anymore; in addition, the office computer is no longer nearby (determined by the Bluetooth RSSI). The Web application is migrated, manually, automatically or after user confirmation (depending on whether the test mode is automatic or assisted), from the computer to the phone. User confirmation is requested, in assisted modality, from the phone (an alert belonging to the Context Monitor pops up and keeps in foreground until the user accepts/rejects the suggested migration trigger). This way, the workflow is not interrupted by the transition from the office to the meeting room.

In this scenario, differently from the previous one, migration can be both pushed or pulled. Indeed, in the manual mode migration is pushed from the source to the target device, while in the automatic and assisted mode the flow is conceptually different. In these two cases, the Context Monitor asks the Platform to migrate the page from the desktop device to the smartphone itself. Thus, a pulling is implicitly originated in the Context Monitor running on the smartphone.

## 4 Results

In this section we present the results of the studies carried out to explore the user needs and requirements for context-aware information sharing in an MDE. The factors that we were interested in with our study were related to the existence of (1) specific user needs for information sharing and (2) relevant preferences for a specific operational mode of how the user would be willing to interact with the system (e.g. performing actions manually, assisted by the system or letting the system to perform actions automatically). We were also interested in possible relationships between individual user factors, interaction behaviour and declared ratings for the tested system.

In the following subsections we outline the main results of the studies. First, we present the discovered user needs for context-awareness when sharing information between devices. Then we go through the results of the exploratory research done to assess how the actual system performance will affect the perceived usability and experience of the user and what kind of preferences the users have towards the operation of the system.

### 4.1 User needs for context-awareness in multi-device environments

To explore the user needs for context-awareness in MDEs we carried out a longitudinal *diary study*, in which the participants logged their daily needs for information sharing between devices. With the diary study we wanted to probe the most potential tasks, situations and devices for information sharing that people would find the most beneficial during their everyday life. All 24 test participants joined the diary study. All in all, we gathered 91 diary entries during almost 3 months' time spanning from May 2012 to August 2012.

#### *4.1.1 Devices and situations*

Most of the user needs gathered in the study were sequential interactions with devices. With sequential interaction we mean a situation in which the user switches from one device to another while continuing with the on-going task.

In addition to sequential interaction the participants reported needs for simultaneous interactions with the devices, such as automatically streaming child monitor content (audio and video) to a home TV system using *picture-in-picture* functionality.

The information sharing needs found from the study can be divided into two main categories: *object-related* and *identity-related.* The *object-related* information sharing needs include items such as photos, videos and documents. Such information sharing needs are often related to task continuation on another device in a different location (e.g. accessing documents from a work computer at home). The *identity-related* information sharing needs on the other hand are more closely connected to the identity of the user. These needs arise in situations where the user would like to preserve the "signed-in" experience of using a certain application while switching to a different device.

In the study, most common user needs were due to object-related information sharing between a smartphone and a laptop in the home or work environment. As depicted in Fig. 5, most of the needs were about information sharing, such as sharing photos (23 %) and documents (16 %). The discovered needs, especially in the work environment, mostly matched with the traditional user needs. However, significant share of needs were also identity-related, such as migrating the on-going Web session from one device to another.

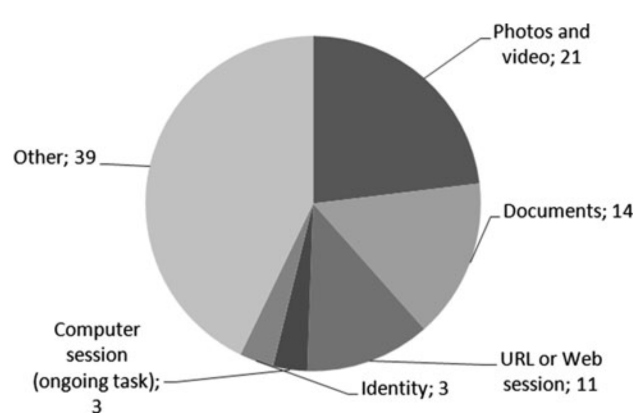



**Fig. 5** Most common information sharing needs between devices as percentages

**Table 1** Source and target devices—information needs between certain devices (percentage (N))

| Source device/target device | Mobile device | Home device (laptop etc.) | Work computer | Other | All |
|---|---|---|---|---|---|
| Mobile device | 4.4 % (4) | 12.1 % (11) | 11 % (10) | 5.5 % (5) | 33 % (30) |
| Home device (laptop etc.) | 12.1 % (11) | 3.3 % (3) | 10 % (9) | 2.2 % (2) | 27 % (25) |
| Work computer | 11 % (10) | 0 % (0) | 11 % (10) | 2.2 % (2) | 24 % (22) |
| Other | 10 % (9) | 1.1 % (1) | 0 % (0) | 4.4 % (4) | 15 % (14) |
| All | 37 % (34) | 16 % (15) | 32 % (29) | 14 % (13) | 100 % (91) |

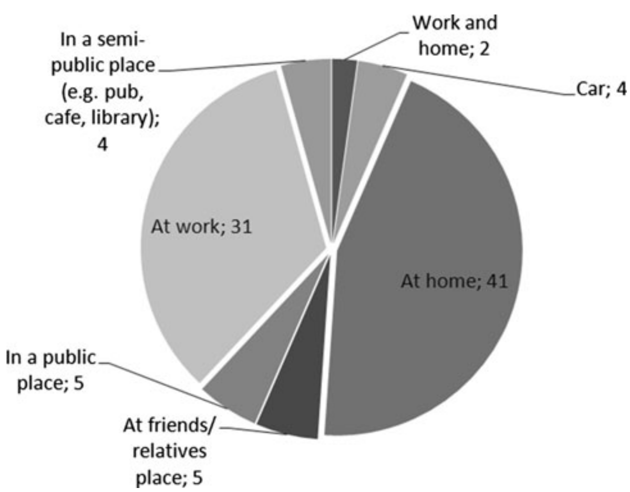



**Fig. 6** The distribution of information sharing needs in different locations as percentages

According to the study (see Table 1), the most common *source devices* (device from which the task would be migrated) were (N = 91): mobile devices (37 %) and work computer (32 %). The need of sharing information from tablet devices was considered only in 4 % of diary entries. The most common *target devices* in the study were: mobile devices (33 %) and home computers (27 %).

The most common places where people had needs to share information between devices (see Fig. 6) were *at home* (45 %) and *at work* (34 %). Other locations were mentioned significantly less: public places (5 %), friends or relatives place (5 %) and places, like cafés or libraries (4 %).

Based on the results, we can see that the information sharing needs *at home* has the most variety with the devices used. On the one hand, there are needs to share content between smartphones, tablets and laptops as well as other devices, such as TVs and other household devices. On the other hand, *at work* people mostly need to share content and applications between different computers (e.g. own and work laptop and a work desktop computer).

As depicted in Table 1, many of the information sharing needs discovered in the study involved mobile devices either as a source or target device for migration (slightly more often as a source device). Based on the study, migrating from a work computer to either a mobile device or to a home computer was also a common need. All in all, mobile devices were involved in more than third of all information sharing needs (37 % as source device and 33 % as target device).

#### 4.1.2 *Analysis of user needs in different situations*

> *"A major difficulty is keeping the data in each device up to date."*—Participant #7, male, 28 years old.

While analysing the user needs from the diary study we grouped them into different themes. One such theme was *data synchronisation*. Data synchronisation is still a major challenge with today's computing systems, even though there are many possibilities. On the other hand, the vast amount of possibilities (email, cloud storage, physical medium etc.) might also make the task mundane. Based on the diary entries, we can say that there is a need for making the data synchronisation task easier, more straight-forward and automated, with less focus on the medium and more on the task itself.

> *"When you use email to send files from a mobile phone to a desktop you have to wait for the time to send the email and for the time to receive it. When you send a file through Bluetooth, the first time you have to authorise the device, and every time you have to turn on the Bluetooth, select the target device and after the information migration, turn off the Bluetooth—it's a slow process."*—Participant #5, male, 30 years old.

Another apparent theme which emerged from the data was *task continuity*. The data could be in sync between all different devices, by e.g. being stored in the *cloud* (i.e. backend Web servers), accessible by all the devices. Nevertheless, the flow of interaction would be still discontinued as the user has to re-open the current application (or a corresponding one) on the new device and even log into a service etc., the task continuity in MDEs is either non-existent or very poor. Thus, in addition to providing ways and means for easily migrating and synchronising data, migration and the sharing of whole applications including UIs, *session and state information* as well as *the current context of use* should be offered by the systems.

> *"Basically, I use my phone mostly at home, for social media, email, news and everything, but the screen is small and it is difficult to write with a touch screen device. It would be good if it would be possible to easily change all my services to another terminal!"*—Participant #20, female, 31 years old.

Going a bit further from technical needs, another theme that emerged from the diary entries was a need to *keep the digital identity in sync*. This need is also discussed by Anderson et al. (see *What We Carry*, within (Schilit and Sengupta 2004)). Contact lists, media libraries, *Facebook*, *Twitter* and other social media credentials etc. also need to be in sync and accessible on several different devices, so that when one changes from device to another, the kind of data that "one's digital identity comprises of" is also present on the other device (Schilit and Sengupta 2004). Today's computers and other digital devices are increasingly linked to Web services, social networks etc. This leads to a situation where there's an increased need for easily switching devices while keeping your personal identity and session credentials. The task is not only technical, but also an issue of interaction design.

> *"Home computers usually have also other family members' user accounts in them (social media, chat, email), so it is always time consuming to change the profile, log in with your own usernames and passwords."*—Participant #20, female, 31 years old.

#### *4.1.3 Preferred interaction in different situations*

According to the study it seems that people would mostly prefer the interaction with the system to be somewhat controlled by themselves. As we can see from Fig. 7 below, for most of the tasks recorded in the diary study people preferred that the migration would be either fully controlled by the user (27 %) or intelligently suggested by the system in suitable/opportune moments (46 %). In 25 % of the cases people would allow the system to perform an automatic migration of task or content from one device to another.

If we look at the suitable interaction methods per location or place, we can see that people have different preferences of allowing the system to do automatic migrations at home rather than in the office, which were the two most common locations according to the study (see Fig. 8). At home people would allow the system to perform more automatic migrations. On the other hand, at work people preferred to have control over the task and the system behaviour. In the three main situations, in public places, at friends' and in a car, it is preferred that the system uses the suggested mode in migration. It is interesting that in those situations an automatic mode is not an option. Figure 8 below depicts the preferred interaction modes per each recorded place or location.

### 4.2 User acceptance of system performance

In this section we present the evaluation of our prototype system to explore how proactive and context-aware UI migration is perceived from a user's point-of-view and which technical requirements are needed for it. The factors that we were interested in with our study were related to the existence of relevant preferences for a specific migration modality (manual, assisted and automatic) and of possible relationships between individual user factors, interaction behaviour and declared ratings for the tested system.

#### *4.2.1 System functionality*

All relevant event times were logged in order to count the latencies of the system. The user tests were also recorded

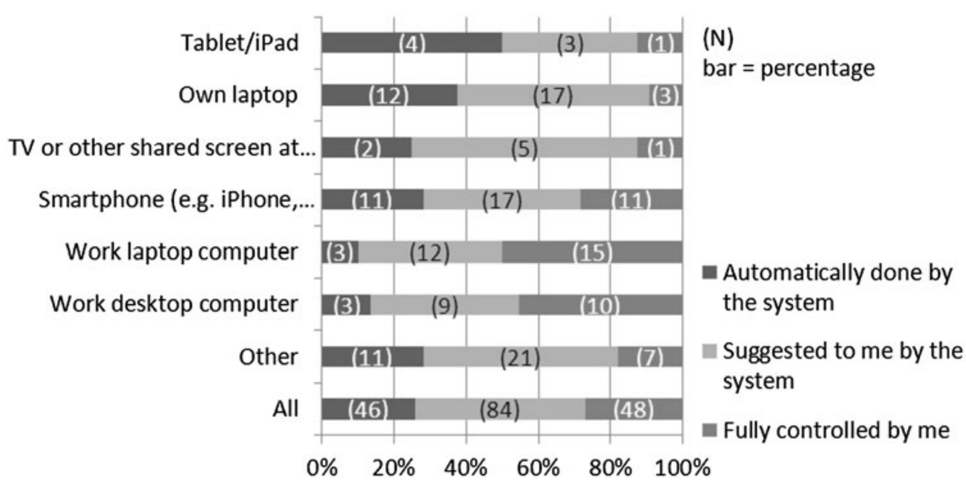



**Fig. 7** Preferred interaction with the system

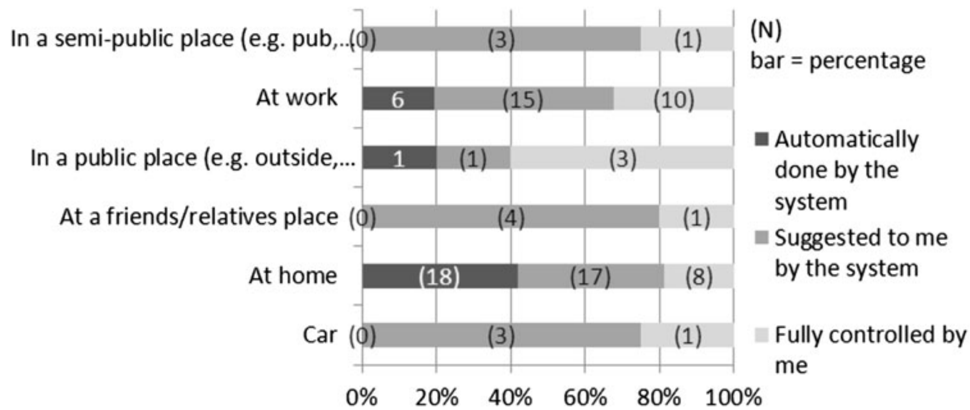



**Fig. 8** Preferred interaction modes per place or location

on video in order to count the users' focus shifts between the devices during each migration mode and to catch comments. The migration modes were shuffled for each test participant, thus obtaining six different orders to minimise any learning effects.

For the focus shifts, the hypothesis was that when using the manual mode for migration, there would be more focus shifts between the devices, as the users would check between the devices, and when moving towards more automatic migration there would be less focus shifts.

### *4.2.2 Quantitative results*

In order to be able to calculate the relevant latencies, we recorded all the relevant events on a server log, such as:

- "mobile device placed on the table",
- "migration to PC triggered on the mobile device",
- "Web page opened on PC",
- "mobile device picked at hand",
- "migration to mobile device triggered on the PC" and
- "Web page opened on mobile device".

For the manual mode, we calculated the latencies between the user triggering migration on the mobile device to when the Web page is opened on the computer (*A*) as well as between the user triggering migration on the computer to when the Web page is opened on the mobile device (*B*). For the assisted and automatic modes, we calculated the latencies between the mobile device being placed on the table to when the Web page is opened on the computer (*C and E*) as well as between the mobile device being picked up and when the Web page is opened on the mobile device (*D and F*). These latencies are depicted respectively in Fig. 9 (baseline in the figure is manual mobile to PC median latency, 2.9 s). As it can be seen, in the assisted and automatic modes there was significantly more latency. This is due to the context detection algorithms that take a few seconds to detect the events of "being placed on the table"/"lifted from the table". Indeed, the algorithm continuously monitors acceleration/tilting and needs some time in order to avoid false positives (unwanted migrations).

In the case of manual trigger, instead, the time behaviour is mainly due to: (a) latency of the page elaboration (client- and server-side) and (b) network delays.

Client-side, the page elaboration consists of the serialisation of the page, i.e. the creation of the *extensible markup language* (XML) string document from the DOM on the browser. Client/side serialisation is done by a JavaScript procedure. Server-side, the elaboration consists of: parsing the XML string to create a document object, filling the document object with additional information (e.g. user interaction state) and serialising it into a text/HTML file that would be accessed by the target device browser.

We argue that, because of the system latencies, assisted and automatic migration led to the highest amount of focus shifts (see Fig. 10). Therefore, we had to claim our initial hypothesis (less focus shifts when moving towards more automatic migration) false for this study. For future work,

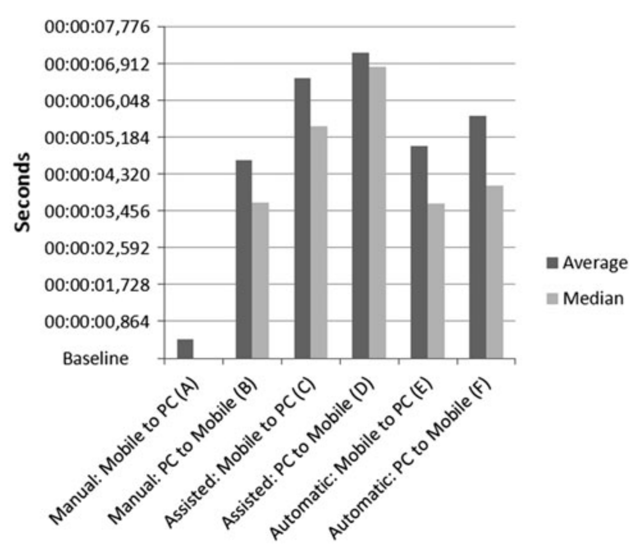



**Fig. 9** Latencies of the different migration modes

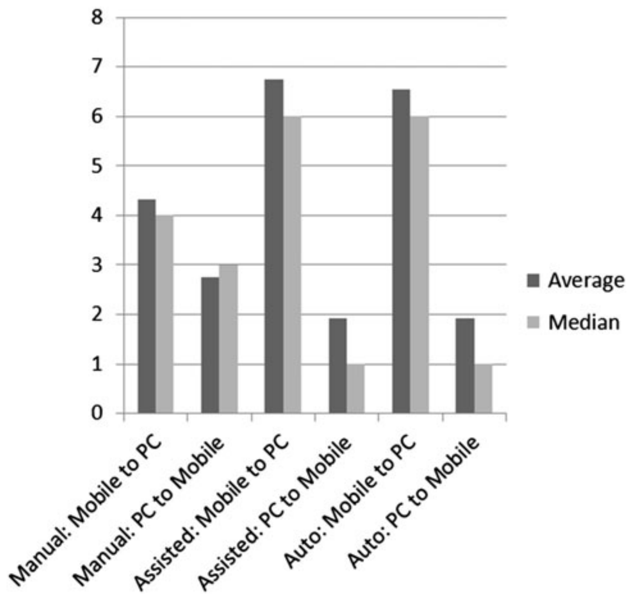



**Fig. 10** Number of focus shifts during the different migration modes

the assisted and automatic mode latencies will be reduced as much as possible. The system, as pointed out by many of the test participants, should also give some sort of hint of the migration being in progress, e.g. by using a small progress bar over the screen.

Nielsen (1993) discusses the acceptable latencies/ response times for any application in regards to usability:

- *0–100 ms* is approximately the delay for the users to feel that the system is reacting instantly, needing no other feedback than just to display the resulting action/ data.
- *100 ms–1 s* is approximately the delay for the users to feel that the flow of use is not interrupted, though the users will notice the slight delay and lose the feeling of direct manipulation. No special feedback is normally required from the system.
- *1–10 s* is approximately the delay for keeping the users' attention. Longer delays will result in the users doing/wanting to do something else whilst waiting for the system to perform. Feedback during the delay is important and required; otherwise the users do not know whether the system is actually doing anything.

Being still a prototype, our current system falls into the last category and needs to have a progress indicator(s) during the migrations. Furthermore, considering the context-detection algorithms and (a) the latency of the page elaboration (client- and server-side) and (b) the network delays, it will be very challenging to develop the system to perform in the time range of 100 ms to 1 s. However, the participants commented on the free-form sections of the questionnaires that the system already performed well, which give us initial direction that a migration latency of few seconds is acceptable, as long as hint(s) of the system progress is displayed. In general, we also gained a good insight into the trade-off between the system latency and the detection quality. The participants commented that a slight latency is tolerable, if the context detection does not produce false negatives/positives and system progress is visualised.

During the laboratory test, for assisted/automatic migration, we recorded the false positives and negatives. The former were unwanted migrations, i.e. triggered when the user did not need them. The latter are migrations that did not occur even if the user believed that all the conditions for assisted/automatic migration were met (i.e. proximity to the target device, etc.). Twelve users faced at least one false positive or negative. The great majority of those unexpected behaviours were false negatives, i.e. 33, while the false positives were only 5. Among such 38 context-detection failures, 36 occurred when migrating from mobile to PC, and only 2 from PC to mobile. These can be argued to be caused by the context detection algorithms not detecting the "arriving to the office" events as reliably as the "leaving the office" events. This further underlines the many problems of activity recognition, which our application does not try to solve, only acting as an enabler for fair comparison between the different migration modes investigated in this study.

### 4.2.3 *Qualitative results*

During the laboratory test, we had the participants fill in an in-between questionnaire after each migration mode and also a post-test questionnaire after the test. Thus, we acquired four questionnaires of the laboratory test per user plus their diary markings to qualitatively evaluate our concept and prototype.

Users rated various aspects of the system by providing a rating according to a 1-to-5 *Likert Scale* (with 5 as most positive and 1 as most negative value).

The in-between questionnaire results are depicted in Figs. 11 and 12.

As it can be seen from the results, there were no major differences between the score of the various migration modes. The slightly better scores in favour of the automatic modes can mostly be assumed to be the effect of the complexity of our early prototype in regards to the manual mode. The manual mode requires some manual interaction by the user (i.e. switching from the visited Web page to the tab/window of the Migration Client, selecting the desired target device among the available ones, triggering the migration). Some of the test participants also commented that the manual migration could be much more usable if the

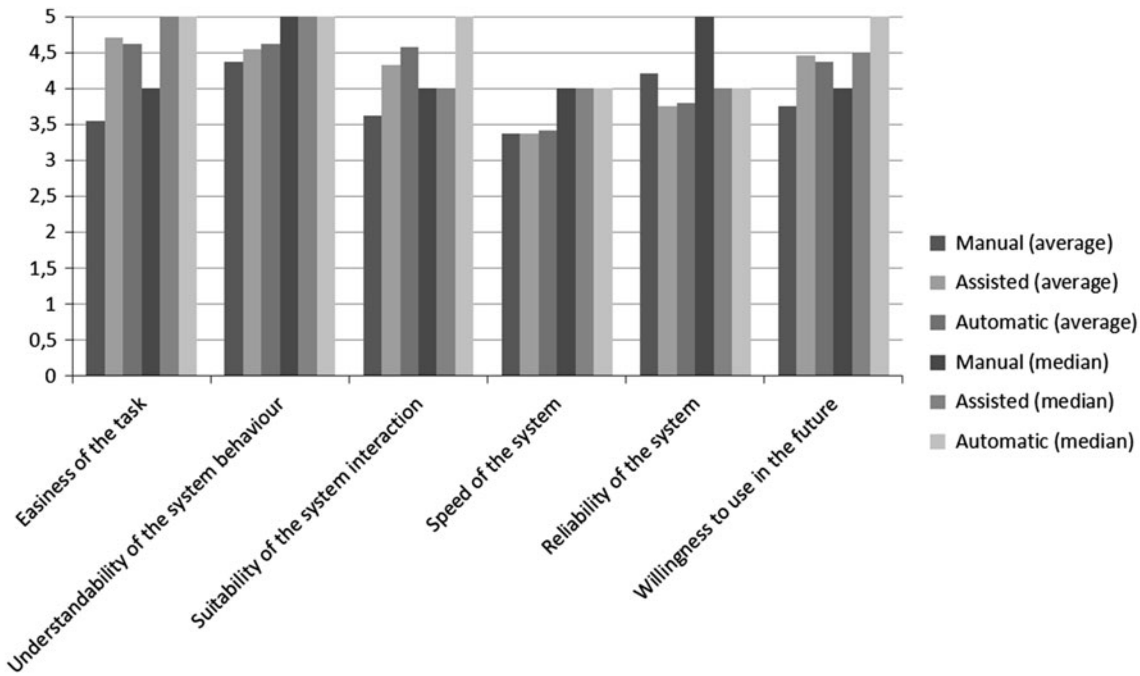
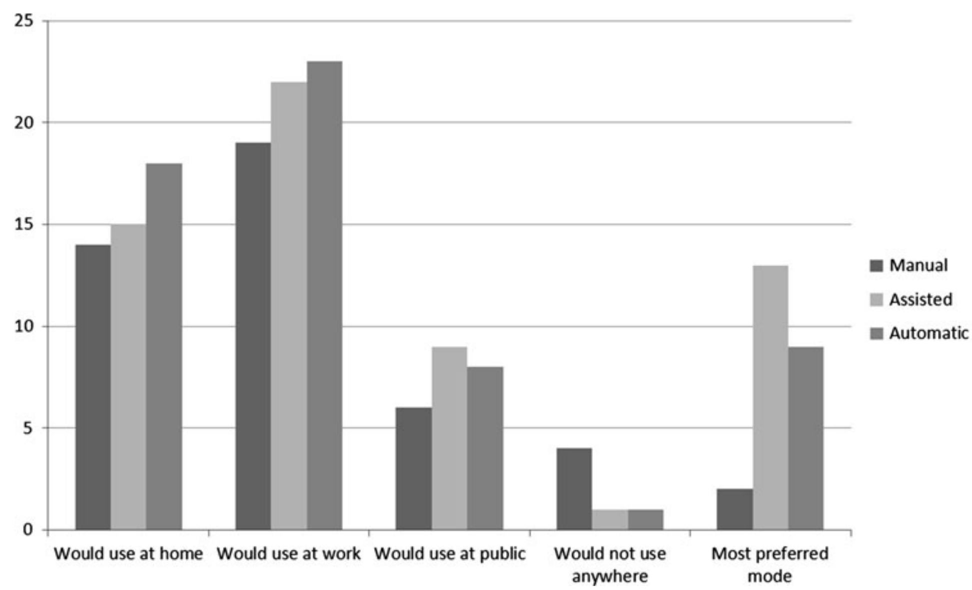



**Fig. 11** Qualitative assessment of the system functionality

**Fig. 12** Preferences and usage of the different migration modes




Migration Client was implemented as an easy-to-use and intuitive Web browser plugin.

Most of the minor differences among the migration modes are related to the easiness, suitability, reliability and the willingness to use the selected mode in the future. As stated before, the difference in easiness favouring more automatic modes is due to the complexity of the current prototype that was perceived as not so easy to use.

Additionally, the suitability favours the automatic mode more, as the task to perform in the user test was quite trivial (migration of a web page from one device to another and back). It can be argued that the suitability of a migration mode is highly dependent on the task to be performed. The test participants also commented that, on a more public setting, they would prefer having full control of the migration, i.e. more manual migration. The reliability of the system was perceived to be better during the manual mode but, as stated before, it can also be argued to be the effect of our current prototype not giving any progress hints of the migration when the assisted and automatic modes were used. The users also commented on this quite a lot. Thus, it is reasonable to assume that there are no major differences

**Table 2** Summary of the Pearson correlation tests

| Variables | Migration task | | | | | |
|---|---|---|---|---|---|---|
| | Manual | | Assisted | | Automatic | |
| | Mobile-PC | PC-Mobile | Mobile-PC | PC-Mobile | Mobile-PC | PC-Mobile |
| Latency, shifts | | | | | | |
| Correlation | 0.151 | −0.042 | −0.052 | −0.108 | 0.125 | −0.116 |
| Significance | 0.48 | 0.845 | 0.809 | 0.617 | 0.561 | 0.588 |
| Shifts, average usability | | | | | | |
| Correlation | −0.228 | −0.344 | −0.066 | 0.016 | 0.215 | −0.361 |
| Significance | 0.285 | 0.1 | 0.76 | 0.941 | 0.313 | 0.083 |
| Latency, average usability | | | | | | |
| Correlation | −0.255 | 0.147 | −0.212 | −0.013 | −0.037 | 0.019 |
| Significance | 0.228 | 0.492 | 0.32 | 0.953 | 0.866 | 0.931 |
| Shifts, speed usability | | | | | | |
| Correlation | −0.187 | −0.256 | −0.054 | 0.196 | −0.041 | −0.168 |
| Significance | 0.383 | 0.228 | 0.802 | 0.359 | 0.851 | 0.433 |
| Latency, speed usability | | | | | | |
| Correlation | −0.351 | −0.083 | −0.216 | 0.133 | −0.54 | 0.231 |
| Significance | 0.092 | 0.699 | 0.311 | 0.535 | 0.006 | 0.277 |
| Shifts, suitability | | | | | | |
| Correlation | 0.321 | −0.131 | 0 | −0.093 | 0.107 | −0.03 |
| Significance | 0.126 | 0.541 | 1 | 0.665 | 0.618 | 0.891 |
| Latency, suitability | | | | | | |
| Correlation | −0.473 | 0.075 | 0.187 | 0.079 | 0.146 | −0.201 |
| Significance | 0.02 | 0.729 | 0.381 | 0.714 | 0.496 | 0.347 |
| Shifts, willingness | | | | | | |
| Correlation | −0.132 | −0.391 | −0.169 | −0.075 | −0.114 | −0.411 |
| Significance | 0.539 | 0.059 | 0.431 | 0.726 | 0.596 | 0.046 |
| Latency, willingness | | | | | | |
| Correlation | 0.01 | 0.189 | 0.056 | −0.179 | 0.081 | 0.258 |
| Significance | 0.961 | 0.376 | 0.795 | 0.402 | 0.707 | 0.223 |

in the migration modes and that the selection of the optimal mode should depend on the task at hand. However, regarding the willingness to use the available modes in the future, the automatic mode was the preferred one. Indeed, it was already seen as quite a ready functionality to be used in more trivial everyday tasks, such as migrating music player controls from a mobile device to a car steering wheel, etc. Overall, the most preferred migration mode was the assisted one, thanks to the combination of simplicity (migration is suggested automatically) and user control (migration has to be explicitly allowed before being executed).

### *4.2.4 Statistical analysis*

Statistical tests have been applied to the data gathered during the trials. The aim was to discover possible correlations between several aspects (i.e. variables), such as between the number of video-detected focus shifts of the user and her/his rating. We ran *Pearson correlation tests* over the system latency and focus shifts in relation to the average usability, system speed (usability), suitability of the interaction mode and the willingness to use the system (and the selected mode) in the future. The Pearson correlation tests are summarised in Table 2.

The following variables have been considered:

- *Latency*: The time needed to perform migration, as previously described in subsection *System functionality*.
- *Shifts*: The number of visual focus shifts between devices. A focus shift occurs anytime the device the user is gazing changes (i.e. from PC to smartphone or from smartphone to PC).
- *Average usability*: The mean value among the following usability-related ratings of the user: task easiness, system behaviour understandability, system interaction

suitability, system speed, system reliability, willingness to use the system in the future.
- *Speed usability*, *Suitability* and *Willingness*: These refer to system speed, system interaction suitability and willingness to use the system in the future, respectively.

The correlation factor, labelled as *Correlation*, is conventionally classified as follows (in absolute value):

- 0: no correlation at all (i.e. independence between the variables);
- Below 0.3: weak correlation;
- Below 1: moderate correlation.

The correlation significance is indicated by *Significance*, which is measured by the following intervals:

- 0–0.01: highly significant;
- Below 0.05: statistically significant.

It is worth pointing out that the aim of these tests is mainly exploratory, because they are not actually devoted to accept/reject any predefined hypothesis. Therefore, even if multiple comparison tests involving the same variables were performed, no correction has been applied to the significance intervals.

Such tests show the following statistically interesting aspects:

1. There is highly significant (negative) correlation between latency and system speed usability when using automatic mode from mobile to PC.
2. A statistically significant (negative) correlation exists between latency and suitability when manually migrating from mobile to PC.
3. Another statistically significant (negative) correlation for the automatic mode was found between focus shifts and willingness to use the system in the PC-to-mobile task.
4. A moderate (negative) correlation, although not statistically relevant, was found between latency and system speed usability when manually migrating from mobile to PC.

Given these findings, we can argue that they support our notions on how the system was perceived during the tests. Automatic migration took a longer time (due to the context-detection algorithms delay) when migrating from mobile to PC and clearly had the participants question themselves about system usability. Also, when using the manual mode in migrating from mobile to PC, the latency reduced the interaction suitability and had the participants commenting on the manual mode usability.

It is also worthwhile to mention that the statistical results neither show direct correlation between latency and focus shifts, nor between these two variables and the average usability. This lack of correlation is indicated by the low correlation factor (weak, in most cases) as well as by the significance value, which is much bigger than 0.05. We could argue that the system already performs "good enough" to provide such context-aware migration functionality. As mentioned before, this also shows in the participants' free-form comments that, overall, they already found the system to be quite good.

We also ran statistical difference tests on the ratings (Likert scale 1–5) the participants gave during the laboratory tests and the system latencies and found out several significant differences. The null hypothesis on the tests was H0: "The modes perform similarly" (i.e. the means of the groups are the same).

The results show that we have to reject our null hypothesis (using the conventional 0.01 and 0.05 alphas) on the following cases of the system functionality assessment:

a) Task easiness, assisted vs. manual, highly significant (*p* value 0.0001)
b) Task easiness, auto vs. manual, highly significant (p value 0.0008)
c) Interaction suitability, assisted vs. manual, statistically significant (p value 0.0233)
d) Interaction suitability, auto vs. manual, highly significant (p value 0.0059)
e) Willingness to use in the future, assisted vs. manual, highly significant (p value 0.0093)
f) Willingness to use in the future, auto vs. manual, statistically significant (p value 0.0281)
g) Latency, mobile-to-PC, assisted vs. manual, highly significant (p value $<0.0001$)
h) Latency, PC-to-mobile, assisted vs. manual, statistically significant (p value 0.0284)
i) Latency, mobile-to-PC, auto vs. manual, highly significant (p value $<0.0001$)

The aforementioned preliminary statistics confirm that the preference towards automatic and assisted modes (with respect to the manual mode) is considerably relevant.

These statistics are related to confirmatory data analysis, as they are devoted to reject the null hypothesis. *Bonferroni Correction* has thus been applied with significance level ($\alpha/2$), aiming to reduce possible family-wise errors. Some variables (i.e. latency in mobile-to-PC manual mode and task easiness manual mode) were indeed involved in two comparisons. The correction lowers the alphas from 0.01 and 0.05 to 0.005 and 0.025, respectively. If such corrected alphas are considered, then some downgrades in the statistical significance occur: (d) and (e) are re-classified from highly significant to statistically significant, while (f) and (h) change from statistically significant to insignificant. Nevertheless, even after applying Bonferroni correction, most comparisons are still statistically meaningful. For instance, the e) corrected comparison between the willingness to use assisted and manual modes, is also significant, which is interesting as it suggests a preference of the users towards the assisted mode.

Based on these results, we can argue that the assisted and automatic modes were easier to use in the test and suited the given interaction situation better. The participants were also keener to use the assisted and automatic modes in the future vs. using the manual mode. However, it is worth mentioning that, given further development to the system by making the manual mode more easier, the users commented that the manual mode could be useful in some cases as well (e.g. in public spaces when definite explicit control is more desirable).

### 4.3 Requirements for context-awareness in multi-device environments

The evaluation we carried out, both as a diary study and as laboratory tests, led us to a series of results that can be grouped into three main categories:

- *Potential use cases*—through the diary study we investigated the typical tasks to be migrated and contexts where migration would occur more often;
- *General usability indications*—through the descriptive statistics gathered from the laboratory test results and the informal feedback from the participants' we analysed the preferred migration modality and the perceived usability aspects (i.e. criticisms and recommendations);
- *Concrete usability indications*—through the confirmatory statistics applied to the data collected during the laboratory test we investigated the correlations between different test parameters (e.g. number of user focus shifts on some modality and their willingness to use that modality in the future) and significance of difference between ratings (e.g. task easiness of assisted vs. manual mode).

By analysing these findings, we drew an initial list of requirements for optimal context-awareness in MDEs, together with required actions to meet them.

- **R1**: *Support a diverse set of devices*
  - *Finding:* Users have diverse needs in sharing information between devices (see the diary study) on migrating across diverse devices: tablets, phones, PCs.
  - *Guideline:* The system should be compatible with a diverse set of devices (cross-platform compatibility).
  - *Requirement:* The client-side of any assisting migration support should be developed as OS-independent (as in our case with a Web based approach), or providing an implementation for each OS. In the latter case, cross-platform functionality should be ensured.
- **R2**: *Adapt the system behaviour to the current situation*
  - *Finding:* Users often have different migration needs in different situations: more control (manual mode) is preferred in public environments and more automation on private contexts.
  - *Guideline*: The system should be able to adapt the operational behaviour according to the current situation (home, work, public place).
  - *Requirement:* Context-awareness should be used to provide assistance in situations where the user is willing to give control over to the migration as well as give back the control in situations where full control by the user is required (e.g. at work or public places).
- **R3**: *Minimise latency and give feedback when performing automatic functionality*
  - *Finding*: System latency is considered to affect the perceived speed (thus impacting on usability) at least in the automatic mode.
  - *Guideline:* Maximise system responsiveness and give feedback to the user when possible.
  - *Requirement*: Optimise the algorithms for context sensing and for migration processing in order to reduce the delay.
- **R4**: *Provide easy navigation and one-step interaction for manual operation of the system*
  - *Finding:* Latency affects the perceived suitability (thus impacting on usability) in manual mode.
  - *Guideline:* Simplify the mechanism for manually trigger migration
  - *Requirement:* Optimise the usability of the control panel, e.g. by providing a one-step interaction for triggering migration from one device to another.

## 5 Discussion

By considering recent technological proposals previously mentioned, such as Google Chrome synchronisation capability, MOVL Connect Platform and Pocket, similarities with our system can be found. Such proposals are also aimed to let the user seamlessly move the user interface or data in use across (heterogeneous) devices, but differ technically from ours. In our system, the only piece of software that needs to run on the (mobile) client device to allow migration is the Context Monitor. The remaining modules of the platform are totally Web-based, and thus do not require any plugin or additional application but a Web browser running on the device. With our Web-based strategy, a generic stationary device could be involved in migration without the need to install a specific application (such as Google Chrome browser or Pocket) or re-developing the application with a specific library (such as in the case of MOVL Connect Platform). For instance, a TV with embedded Web browser would be suitable for acting as a source or target for our context-aware migration, as only the mobile device would need to host the dedicated Context Monitor.

In addition, the abovementioned solutions do not in any way monitor the context of the user and therefore cannot migrate the UI automatically or suggest migration based on contextual factors. As mentioned, the main challenge for us is to support seamless device changes and by providing intelligent, situation-aware functionality this challenge can be addressed. As more and more devices are being equipped with different kinds of sensors, they will be able to know more and more context information about themselves, such as location/presence of other suitable targets for UI migration.

According to R1, the system should support diverse devices: tablets, phones, PCs. The highest compatibility can be given by a Web-based implementation.

R2 suggests system adaptation to current context/situation (e.g. private vs. public situations and automatic vs. assisted migration) in addition to "plain technical adaptation" (i.e. adapt screen/interface size when switching from PC to mobile etc.). The former capability should assure optimal privacy by protecting data migrated across devices (an example is the user willing to have full automatic migration across domestic devices, while preferring the assisted mode in public areas). The latter is an important factor in addition to content and navigation parity, which means that the content/functionality should be preserved along the migration (e.g. using a mobile device to search for flights should not produce different results when performed on a PC) and that, while screen/interface size should be adapted to the target device properties, navigation should still be as consistent as possible. However, as mentioned in the related work section, task continuity should be prioritised over full UI consistency, thus summarising R2 as intelligent adaptation to device and surroundings context, where the consistency is "good enough" and task continuity is best supported.

R3 could be seen as a self-evident requirement, but in this case, where the system is based on Web (i.e. network connections and their latency) as well as activity/context detection algorithms, this requirement is an important one to consider when designing such systems. When designing and implementing such systems it needs to be considered what is the acceptable amount of latency and the trade-off of latency vs. false negative/positive activity/context detections.

R4 instead highlights the need for simplifying the mechanism for manual migration trigger, for instance by providing a device-dependent implementation of the migration control panel (Migration Client). Additional modules (to be deployed on the client) could be developed to further improve UX: a plugin or a dedicated application may be more usable than a Web interface for triggering manual migration in some situations. Nevertheless, in order to avoid possible conflicts with R1, it is worth noting that such modules would not replace the main Web-based architecture.

We believe that, besides the possibility of extending our platform with modules implemented ad hoc, our approach can provide a reasonable trade-off between flexibility and usability.

The main limitation of the proposed prototype is that, having been specifically designed for the Web, it is not suitable for migrating native applications. Another drawback is due to the proxy-based strategy, which increases navigation latency. Optimisations, even though out of the scope of this study, are possible. An improvement in performance could be achieved by a caching strategy in the proxy. A reasonable solution for speeding-up navigation could be caching the annotated Web resources (e.g. HTML pages).

## 6 Conclusions

In this paper, an integrated solution for automatic (implicit) and assisted (suggested) context-aware migration of Web application has been presented and studied, by comparing it with a manual (explicit) one.

On the one side, the focus has been put on relationships between data technically gathered through the laboratory test and users' feedback. On the other side, user needs have been investigated during a diary study.

Interesting correlations between, for instance, system response time and more subjective aspects, such as suitability and perceived reactivity of the system, were discovered through the laboratory tests. Statistically relevant preferences for automatic and assisted migration modalities with respect to the conventional manual triggering have also been highlighted.

It can be said, based on the Pearson correlation tests, that (1) the average usability of the context-aware migration functionality was perceived to be quite good by the test participants, even though there was latency in the assisted and automatic modes, and (2) more visual cues of the system progress need to be given when the migration is in progress. We can also argue, based on the statistical difference tests on the users' ratings, that the assisted and automatic modes performed better and that the users would prefer those over explicit migration.

The diary study let us gain information about common user needs (i.e. typical use cases) about multi-device interaction (e.g. kind of information and potential types of devices to be involved in migration). By analysing such use cases and considering them jointly with feedback given by the users about our system, we could finally gather some requirements and guidelines. Such findings represent a valuable starting point for improving UX of multi-device context-aware environments.

We can argue, according to the study results, that migratory functionalities for UI and data across different kinds of devices are judged as promising by the users.

The migration mode (manual, assisted or automatic) is highly dependent on the task at hand and also impacts on how users might perceive technical parameters such as system latency.

We conclude that, if context-based migratory functionalities were available and properly tuned, the great public would very likely use them.

**Acknowledgments** The support from the *European Union ARTEMIS Smarcos project* (2012) is gratefully acknowledged.

## References

Adomavicius G, Tuzhilin A (2011) Context-aware recommender systems. Recommender Systems Handbook. Springer, New York, pp 217–253

Ballendat T, Marquardt N, Greenberg S (2010) Proxemic interaction: designing for a proximity and orientation-aware environment. In: Proc. ITS'10, pp 121–130

Brdiczka O (2010) Integral framework for acquiring and evolving situations in smart environments. J Ambient Intell Smart Environ 2(2):91–108

Cheverst K, Dix A, Fitton D, Kray C, Rouncefield M, Sas C, Saslis-Lagoudakis G, Sheridan JG (2005) Exploring bluetooth-based mobile phone interaction with the Hermes photo display. In: Proceedings of MobileHCI 2005, pp 47–54

Dearman D, Pierce JS (2008) It's on my other computer! Computing with multiple devices. In: Proceedings of CHI 2008, pp 1144–1153

Dees W (2011) Usability of nomadic user interfaces. In: Proc. HCII 2011, pp 195–204

Doryab A, Togelius J, Bardram JE (2012) Activity-aware recommendation for collaborative work in operating rooms. In: Proc. IUI 2012, pp 301–304

Fischer C, Gellersen H, Gostner R, Guinard D, Kortuem G, Kray C, Rukzio E, Strengin S (2009) Supporting device discovery and spontaneous interaction with spatial references. Pers Ubiquitous Comput 13(4):255–264

Forrester Research (2013) The European tablet landscape. A technographics data essentials document. February 2013. http://www.forrester.com/European+Technographics+Consumer+Technology+Online+Survey+Q4+2012/-/E-SUS1711#/The+European+Tablet+Landscape/quickscan/-/E-RES91561. Accessed 16 May 2013

Ghiani G, Isoni L, Paternò F (2012a) Security in migratory interactive web applications. In: Proc. MUM 2012

Ghiani G, Paternò F, Santoro C (2012b) Push and pull of web user interfaces in multi-device environments. In: Proc. AVI 2012, pp 10–17

Google (2012) The new multi-screen world: understanding cross-platform consumer behaviour. Research Report, August 2012. http://services.google.com/fh/files/misc/multiscreenworld_final.pdf. Accessed 16 May 2013

Greenberg S, Marquardt N, Ballendat T, Diaz-Marino R, Wang M (2011) Proxemic interactions: the new ubicomp? Interactions 18:42–50

Holleis P, Otto F, Hußmann H, Schmidt (2007) A keystroke-level model for advanced mobile phone interaction. In: Proc. CHI 2007, pp 1505–1514

Könönen V, Pääkkönen P (2011) Optimizing power consumption of always-on applications based on timer alignment. In: Proc. COMSNETS 2011

Marquardt N, Diaz-Marino R, Boring S, Greenberg S (2011) The proximity toolkit: prototyping proxemic interactions in ubiquitous computing ecologies. In: Proc. UIST 2011, pp 315–326

Microsoft (2013) Cross-screen engagement: multi-screen pathways reveal new opportunities for marketers to reach and engage consumers. Microsoft Advertising, Flamingo & Ipsos OTX, USA

Nielsen J (1993) Usability engineering. Morgan Kaufman Publishers Inc., San Francisco

Pyla PS, Tungare M, Pérez-Quiñones MA (2006) Multiple user interfaces: why consistency is not everything, and seamless task migration is key. In: Proceedings of CHI 2006

Rekimoto J, Ayatsuka Y, Kohno M, Oba H (2003) Proximal Interactions: A Direct Manipulation Technique for Wireless Networking. In: Proc. Interact 2003, pp 511–518

Schilit BN, Sengupta U (2004) Device ensembles. IEEE. Computer 37:56–64

Schilit BN, Adams N, Want R (1994) Context-aware computing applications. In: Proc. WMCSA 1994, pp 85–90

Schmidt A (2000) Implicit human computer interaction through context. Pers Ubiquit Comput 4(2/3):191–199

Sellen A, Kurtenbach G, Buxton W (1992) The prevention of mode errors through sensory feedback. Human Comput Interact 7(2):141–164

Smarcos Project (2012) http://www.smarcos-project.eu/. Accessed 16 May 2013

Wäljas M, Segerståhl K, Väänänen-Vainio-Mattila K, Oinas-Kukkonen H (2010) Cross-platform service user experience: a field study and an initial framework. In: Proc. MobileHCI 2010, pp 219–228