Future Generation Computer Systems 34 (2014) 190–200

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Clairvoyance: A framework to integrate shared displays and mobile computing devices

Christian Berkhoff[a], Sergio F. Ochoa[a,*], José A. Pino[a], Jesus Favela[b], Jonice Oliveira[c], Luis A. Guerrero[d]

[a] Computer Science Department, Universidad de Chile, Santiago, Chile

[b] Computer Science Department, CICESE, Ensenada, Mexico

[c] Computer Department, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil

[d] CITIC Research Group, School of Computer Science and Informatics, Universidad de Costa Rica, Costa Rica

ARTICLE INFO

Article history:
Received 28 February 2013
Received in revised form
27 September 2013
Accepted 4 October 2013
Available online 18 October 2013

Keywords:
Shared displays
Information sharing
Encounters
Informal meetings
Peer-to-peer interactions
Ubiquitous computing
Mobile ad hoc networks

ABSTRACT

Supporting formal and informal meetings with digital information and ubiquitous software systems every day becomes increasingly mandatory. These meetings require that the integration of devices participating in the meeting and the information flow among them should be done as seamless as possible to avoid jeopardizing the natural interactions among participants. Trying to contribute to address such a challenge, this article presents a framework that allows devices integration and smooth information flow. This framework, named Clairvoyance, particularly integrates mobile computing devices and large-screen TVs through a mobile ad hoc network, and thus it eases the implementation of shared displays intended to be used in formal and informal meetings. Clairvoyance provides a set of services through an API, which can be used to develop ubiquitous applications that support meetings in particular scenarios. The preliminary evaluation of this framework considered its usage to implement a ubiquitous system that supports social meetings among friends or relatives. According to developers, the framework is easy to use and it provided all required services for such an application. The solution obtained was then utilized by end-users in simulated meetings. The evaluation results indicate that the Clairvoyance services were suitable to support the informal meetings, and that the devices integration and information flow were transparent for the end-users.

1. Introduction

Everyday our physical environments become smarter and more interconnected as a way to improve the services that they provide to people and organizations that use them. As Weiser pointed out [1], the challenge to move our software solutions towards a ubiquitous computing scenario is much more than instrumenting our environments or allowing interactions among these components. Dealing with that challenge also requires, for instance, a smooth and transparent integration among all devices participating in the ubiquitous solution.

The recent widespread availability of mobile devices (such as smartphones, slates and laptops) has energized the development of novel ubiquitous applications that try to take advantage of both, the mobility of these devices and the services eventually provided by the environments where the users are located. Although these devices have improved their computing and storage capacity and also their power autonomy, their screen size is still a limitation to support several regular activities, for instance, formal and informal meetings where supporting information needs to be shared by multiple participants (Fig. 1). In this type of activity, counting on a shared display where to deploy the supporting information has been recognized as mandatory [2–5].

Formal meetings are typically conducted in special rooms equipped with projection capabilities that allow participants to deploy the supporting information on a shared display. These rooms have shown to be useful and effective; however, they are not always available when required and their cost is usually high.

In case of informal meetings, the involved people meet almost anywhere; typically in the place where they encounter or close to that. The supporting information in an informal meeting is usually visualized on the screen of a participant's device, or it can also be shown on a shared display if the people connect such a device to a projector or large-screen TV (LSTV). In most cases, the first option is deficient due to the lack of a synchronous and on-demand access to the shared information for all participants, and also because of

[*] Corresponding author. Tel.: +56 22 9784879; fax: +56 22 6895531.
E-mail addresses: cberkhoff@dcc.uchile.cl (C. Berkhoff), sochoa@dcc.uchile.cl (S.F. Ochoa), jpino@dcc.uchile.cl (J.A. Pino), favela@cicese.mx (J. Favela), jonice@ufrj.br (J. Oliveira), luis.guerrero@ecci.ucr.ac.cr (L.A. Guerrero).

0167-739X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.future.2013.10.013

Fig. 1. Formal and informal meetings where people use a mobile computing device to share information.

its limited capability to show detailed information in a comprehensive way (Fig. 1) [6,7,5].

The second option to display the shared information implies that a person must physically get the projection equipment and connect/disconnect it with cables to a mobile device. This procedure usually interrupts the normal flow of the meeting, distracts participants and reduces the meeting effectiveness. For that reason, most people prefer to simplify that process and show the supporting information directly on the screen of a mobile computing device, although it could be not the best support for those meetings.

It is well-known that informal meetings play an important role for successful collaboration in several settings, such as software development [8], healthcare [9] and education [10]. The dynamics of these meetings is relaxed; therefore, the efficiency in the use of time is not an issue for the participants. By its nature these meetings have no predefined schedule or place of encounter. Yet, these encounters are grounded on awareness of the work environment [11] and often they involve the sharing or exchange of documents. For instance, a study on informal communication in hospitals established that almost half of the informal interactions triggered by physicians included information share or exchange [9]. Social encounters with family and friends can also benefit from the ability to share digital information. Such an event may be an encounter of grandsons and grandparents to look at pictures of the last vacation, or a group of friends meeting to plan a fishing trip for the next weekend. In many of these informal meetings and social encounters the participants are located in places (including homes) that have a LSTV; however, people are not able to take advantage of it due to the burden involved in connecting and disconnecting devices. Finding an available LSTV also represents a limitation for using these large displays, particularly when the participants are not familiar with the surrounding environment.

In this paper we present a framework that allows a smooth integration of shared displays (e.g. a LSTV) and mobile devices. The framework, named Clairvoyance, is based on the authors' previous work [12]. It also provides a service to detect shared displays in the vicinity and it shows information on them. This framework implements an API (Application Programming Interface) that was particularly designed to ease the development of ubiquitous applications that support informal meetings. Since the devices integration is based on a WiFi communication channel, people do not need to use cables or physically manipulate the LSTV device. Thus, the Clairvoyance-based applications can enable any room with a LSTV to become a potential meeting room with capabilities to project supporting information.

The usability and usefulness of the framework was evaluated by software developers, and also by end-users in a real scenario. Although the obtained results are still preliminary, they indicate the framework is easy to use by ubiquitous applications developers and useful to them. The results also show the services provided by the framework are suitable to support informal meetings according to the end-users.

Section 2 presents the related work. Section 3 describes the Clairvoyance framework, including its architecture and main components. Section 4 presents the evaluation scenarios and discusses the results obtained. Section 5 presents the conclusions and future work.

2. Related work

LSTVs and HD projectors are becoming ubiquitous. They are usually available in meeting rooms and increasingly so in corridors, near elevators and other office spaces. LSTVs can also be installed outside the meeting room displaying advertisements or other audiovisual contents. If these devices have some intelligence, they also can support interactions with people [13]. Several researchers have studied the role played by these devices as public screens. In particular, they can be used to provide information to passers-by [14,6], and also as mechanisms to support casual meetings [15,16]. In this direction, Gomez-Goiri et al. [17] proposed to display products locations on a supermarket map, so that disabled customers can decide the best route to reach them. This information is also distributed to the customers' mobile devices.

As LSTVs become ubiquitous and interactive, smartphones may be the input device of choice to interact with them. In a certain sense, smartphones are becoming the universal remote control for several other devices.

Various techniques have been proposed to allow computing devices to interact with LSTVs. However, most of them require considerable effort to set up (e.g. cables or adapters). Jeon et al. [18] present, e.g., several techniques that could be used to deploy images and video/animations on a LSTV, but they do not address the connection process between devices or the related communication issues.

An initial work in this area was SharedNotes which allowed users to create and manipulate notes (text-based messages) between several devices, such as PDAs, PCs and public displays [6]. The system also allowed remote users to participate in a meeting using their workstations.

VNC (Virtual Network Computing) used an approach based on mirroring the contents displayed on a remote screen in a thin client that could be a mobile device, thus allowing the remote, and possibly shared manipulation of the remote device [19]. This mode of interaction supports application sharing through the paradigm of WYSIWIS (What You See is What I See). Although flexible, this approach becomes cumbersome when the difference in size of both displays is significant. In this direction, we can mention the MOVE project [20], which provides transparent access to any machines outside the workplace, offering a uniform and consistent desktop computing environment. The user's desktop computing environment can include customized software, personal data, LSTVs, HD projectors and other utilities, which can be accessed by any computer or smartphone connected to the network, enabling users working anywhere to be as if they were at their own regular workplace without the constraints of mobility and geographical location.

The Pebbles project at Carnegie Mellon University introduced the concept of semantics to support the manipulation of a remote display using a handheld device [21]. Instead of just mirroring the information in both displays, a simple editing interface is made available in the mobile device to interact with the shared display. This concept extends other modes of interaction such as the relatively simple "pick and drop" metaphor that allows information in a mobile device to be displayed on a remote screen [22]. More recently, CGLXTouch has been proposed as an approach to allow multiple collaborators to interact with an ultra-high-resolution display system [23]. In this scenario mobile devices do not render content directly but receive streaming image data from the display system. This contrasts with the approach we propose which aims at moving contents stored in the mobile device to the large display.

A different approach involves the manipulation of a large display through gaze [24], hand gestures or combining both [25]. Gesture Select is an approach that uses mid-air gestures to interact with a wall large display [26]. Cornejo et al. [27] provide a solution that allows users to show pictures in LSTV using hand gestures. Such gestures are captured and recognized by a Kinect sensor which sends particular orders to a computer controlling the LSTV. Although such a solution is interesting, it requires additional infrastructure and for the user to be familiar with the gestures understood by the device. In addition, this solution does not allow the display of contents that is available in the mobile device, without explicitly downloading it to the server controlling the LSTV, which jeopardizes transparency of the sharing process and privacy of the shared resources.

Media streaming allows images, audio and video to be transmitted to a distant device. Kuo et al. [28] propose a simple application-level platform-independent mechanism, which provides seamless media streaming and playing services in a home network. Weinberg [29] developed an approach to make high resolution digital movies using an optical microscope to be shared with a remote audience. Several solutions for media streaming have become commercially available in recent years. Apple AirPlay is a proprietary protocol stack developed by Apple, which allows wireless streaming of media on Apple devices. The use of this technology requires counting on a particular receiver, such as an AppleTV or the airPort Express gateway. We can also mention the integration among LSTVs, HD projectors and other devices in the scenario of Ambient Assisted Living (AAL), fostering the provision of equipment and services for the independent or semi-autonomous living of elderly people within homes and residences. In this direction, López-de-Ipiña et al. [30] described an AAL-enabling platform which combines OSGi middleware, interactive TV, RFID and NFC in order to ease the day-to-day challenges of both elderly people and their care takers and relatives.

Trying to regulate the interaction among these devices the Digital Network Alliance has established the DLNA standard to share multimedia content among several devices, and it certifies devices that adopt this standard. While there is an increasing number of devices that comply with the DLNA standard, the emphasis is on media sharing and not on controlling the remote displays beyond redirecting content, stopping and pausing.

Another interesting proposal is the Wireless Display (WiDi) technology, which has been systematically improved by Intel during the last years. WiDi allows the integration of mobile devices and High Definition TVs (HDTV) using a WiDi adapter [31]. Although this technology is highly promising, the current implementations have two main limitations to address the stated problem: (1) only devices having a WiDi adapter can be connected to a HDTV, and unfortunately such a technology is not massively adopted by the mobile devices manufacturers; and (2) the use of WiDi technology is highly power consuming, which is critical for devices like smartphones.

The integration of a projector within the mobile phone also represents an interesting alternative to address the stated problem. Current devices only support display mirroring, but novel interaction techniques have been explored to deliver commands to the projection device [32]. These interaction techniques include performing pointing gestures in mid-air around the phone, which can be inferred using a camera. These new interaction proposals are mainly focused on addressing the technical aspects of the devices integration; however, the emphasis of our proposal is on utilizing the solution.

3. Clairvoyance framework

In order to exemplify the way in which Clairvoyance works, next we introduce a simulated interaction scenario between a physician and a radiologist, which was adapted from actual observation in a hospital setting, where the inability to share a large display proved disruptive to the encounter. A physician walks out of a patient room in a hospital when he sees the radiologist passing by. He approaches him to ask his opinion about the evolution of a hand injury. The physician displays the most recent X-ray image on his smartphone and they both approach a LSTV located in the nursing pavilion. As they approach the display, the physician transfers the image to the LSTV where they can both discuss the image and analyze the injury evolution (Fig. 2(a)).

Fig. 2(b) shows how the Clairvoyance-based application used by the physician works to allow this integration between the physician's smartphone and the LSTV. The application is able to locate available and busy LSTVs in a certain area by using a Clairvoyance service, as a way to guide interested people towards the closest free shared display; in this case, it is the LSTV located at the nursing pavilion. In order to allow this detection, a Mobile Ad hoc Network (MANET) is automatically implemented by one of the Clairvoyance components. Such a component creates and maintains a MANET that has two types of nodes: the *shared displays* in the area and the *mobile devices* of the people participating in the meeting. The use of shared displays is exclusive for only one device at a certain time instant, and a floor control mechanism is provided for that purpose.

In Fig. 2(b) we can see the MANET has three components: the smartphone of the two meeting participants, and the LSTV. However, just the physician's smartphone is connected to the shared display, and thus it can display its information on the LSTV. When the physician's device disconnects from the LSTV, then the resource will be available to other users. The shared display becomes also available if it disconnects from the mobile device that was using it, or if a timeout for the use of the display (that is configurable) is reached.

Clairvoyance adheres to a client–server architecture; therefore it has a client application running on the mobile device (e.g. a laptop or smartphone) and a server application running on the shared display. If the shared display is a LSTV, the server application runs on a micro-component (a nettop) that is connected to such a device as shown in Fig. 2(b). Counting on a nettop is not required when the shared display is the screen of a laptop or a desktop, since the server module runs directly on that computer.

The client and the server components use the HLMP (High-Level MANET Protocol) API [33] to detect mobile devices and shared displays in the vicinity, and also to automatically form a MANET among them. This MANET infrastructure uses WiFi and includes message routing; therefore it can detect mobile devices and shared displays that are located at more than one hop of distance from the device which is sensing the environment. However, for security reasons, the information transfer between a mobile device and a shared display can only be done when they are located at one hop of distance. Thus we try to avoid that remote users make malicious use of the shared displays. Other possible way to address this security challenge is to use a NFC-based solution, as the one described in [34], which allows a LSTV to identify the participants of the meeting. Only identified people should be able to deploy information on the large screen. Fig. 3 shows the client–server architecture of the Clairvoyance framework, which is composed of four

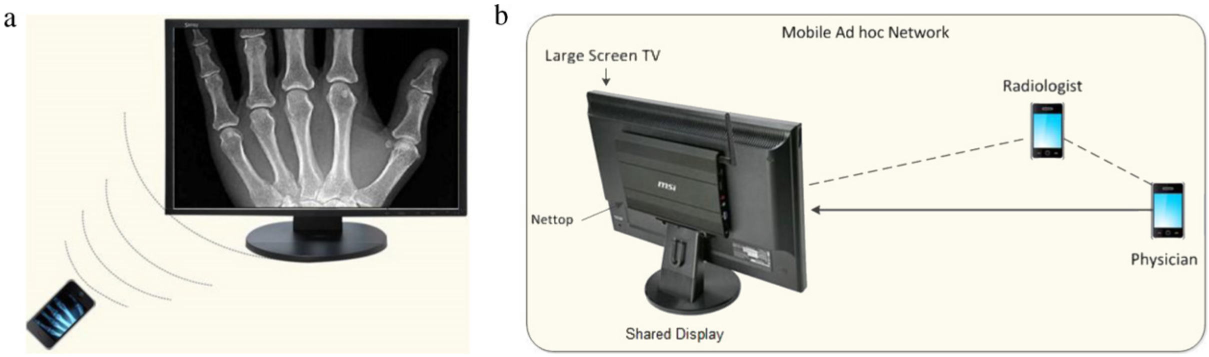


Fig. 2. Clairvoyance interaction scenario.

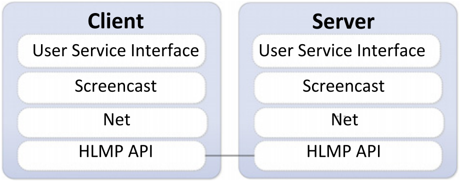


Fig. 3. Basic architecture of the Clairvoyance framework.

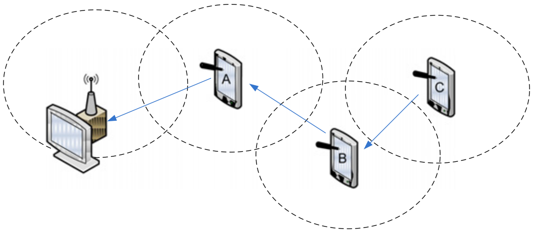


Fig. 4. Example of a MANET implemented with the HLMP API.

layers: the *HLMP API*, the *Net* component, the *Screencast* and the *User Service Interface*.

It is not necessary to use a MANET to support this interaction process. Many peer-to-peer infrastructures, which implement simple and lightweight services, could be used to communicate the shared displays and the client units. In our case, we decided to use HLMP API because the platform features and capabilities were well-known for the authors and we had the capability to adjust the communication services in case of need.

This architecture implements the separation of concerns proposed by Rodriguez-Covili et al. [35]. The lower layer takes care of the communication among the participating devices. The two next layers (i.e. Net and Screencast) provide the coordination services required to support the informal meetings. The upper layer composes services of the lower layers, and thus it provides other services that are ad hoc for supporting meetings and also easy to use by software developers. The services, known as "user services", are typically used by the participant to interact with a shared display and collaborate with other participants (e.g. to share files among them).

The Clairvoyance services have also been classified according to the side of the architecture in which they can be used; therefore we have client services, server services and also services that can run in both sides. Next we present a more detailed description of the four layers composing the framework architecture.

3.1. High-Level MANET Protocol API

As mentioned above, the lower layer uses the HLMP API infrastructure [33] that provides communication support to the devices participating in a meeting (i.e. shared displays and also smartphones and laptops), which are close in a physical environment. This communication support is provided through a MANET. The HLMP API allows message passing among devices and also manages connections, disconnections and identification of nodes in a transparent way for the end-users. This infrastructure is already implemented on WiFi, but it could also use NFC technology. The Clairvoyance communication infrastructure represents a complement to support high-level interactions among the participating devices.

Since HLMP API implements message routing, the MANET can be formed by devices that are located at more than one hop of distance (Fig. 4). However, the routing capabilities are only used to sense the environment (i.e. detecting other nodes in the vicinity) and keep the network topology.

The message delivery strategies supported by HLMP are unicast, multicast and broadcast. The MANET nodes can interact among them using one of the three supported interaction styles: attended, partially unattended and unattended. Attended interactions require the intervention of the users engaged in a meeting. Partially unattended interactions occur when a user interacts with one or more devices, but without interacting with the users of those devices. In that scenario, just the user triggering the interaction request is aware of this process; e.g. when the user explicitly tries to connect his device to a shared display.

Finally, unattended interactions occur when users are not aware of the actions that are being performed by their devices. Unattended interactions are used to implement several awareness mechanisms that are usually embedded in ubiquitous systems; e.g. detection of users' presence, location and availability.

The HLMP infrastructure also provides Clairvoyance awareness information about the users connected to the MANET and file transfer capabilities among their devices, which are interesting services to support informal meetings.

3.2. Net

The *net* component is responsible for managing the logic connection between a client device and a shared display, i.e., it implements a session link. As shown in Fig. 3, the *net* component uses the HLMP API as communication support among the devices participating in a meeting. The *net* services are provided through both, a client and a server module. From the client side, the *CommunicationController* interface allows access to the services provided by the class with the same name (Fig. 5). That class is in charge of configuring and managing the communication link between a client device and a LSTV using the *Protocol* class. The *Constants* class is used by the *Protocol* class to identify the devices participating in the meeting session.

The *CommunicationController* uses the *ScreenEmitter* class, which allows a Clairvoyance client to build the messages containing the graphical information (i.e. *ScreenMessage*). In order to

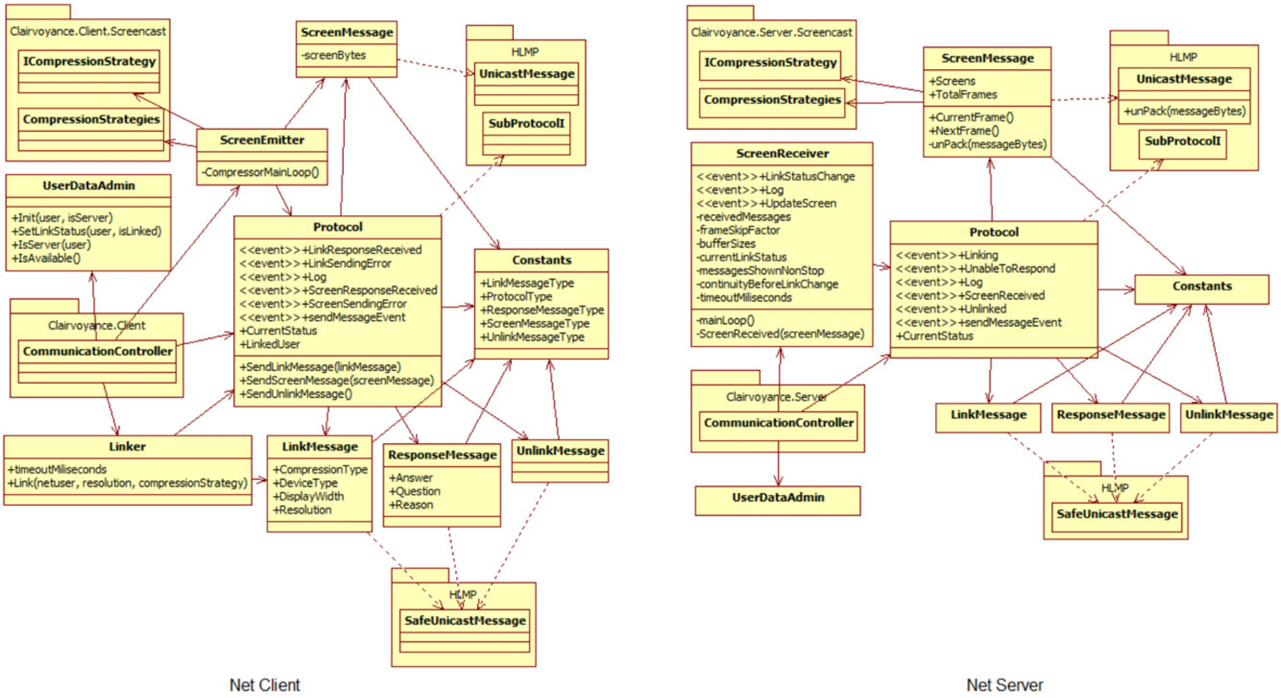


Fig. 5. Structure of the *net* component.

perform this operation the *ScreenEmitter* uses interfaces to the *ICompressionStrategy* and *CompresssionStrategies* classes that belong to the *Screencast* component.

The *net* client module also implements the interfaces to access the services and message types provided by HLMP API, which implements the MANET with the devices participating in the meeting. Each of these devices shares control information with the rest of the nodes by using the *UserDataAdmin* class. The shared information includes the type of participating device (e.g. smartphone, slate, laptop or LSTV) and its status (e.g. available or busy). Thus, it is possible to identify shared displays in the vicinity, and also the distance (in terms of hops) between a client and a server device.

The *Linker* class encapsulates the task of linking a client device and a shared display. In the client side this class implements a blocking call to the server module, which returns a Boolean value. *True* indicates that the linking process was successful, and *false* means that it failed (typically by the server unavailability or timeout). In order to perform this task the *Linker* uses the *Protocol* and *LinkMessage* classes.

The *LinkMessage*, *UnlinkMessage* and *ResponseMessage* message types implement the *SafeUnicastMessage* interface, since they require a more reliable delivery than the one used for regular messages. From the server side the *net* structure is similar to that of the client. However, the server by default is passive and it just reacts to the requests sent by the client devices.

Fig. 6 shows a state transition diagram for a client and a server unit during an informal meeting, and also the interaction protocol (dashed lines). Once the MANET is connected, both units remain idle, waiting for an action from the user. In that case, the user (i.e. client application) can sense the environment, ask for the state or request a connection to a shared display. The server response will depend on the request. Particularly if it is a "connection request" and the state of the display is "available", then the server creates a synchronous connection with the client unit. The client deploys its screen information on the LSTV using this link. That activity is automatically performed until the client decides to disconnect, and in that case both units go back to the initial state. The server implements two threads that allow responding to the client requests, even if it is also connected to a client unit.

3.3. Screencast

This component is in charge of managing the exchange of supporting information between a client device and a shared display. The *screencast* client is in charge of compressing and packing the user interface of the client device connected to a LSTV (or any other type of shared display), and the server module is the counterpart; i.e. it is in charge of unpacking and decompressing such information when it is received. The server module makes this information available to the upper layer component to allow ubiquitous applications to make use of it and deploy it on the screen whenever it is required.

From the client side, the *ICompressionStrategy* class is in charge of selecting one of the two compression strategies currently supported in Clairvoyance: simple or BDS compression. Fig. 7 shows the classes that are responsible of performing that compression (i.e. *SimpleCompressor* and *BSDcompressor*). The information to be compressed is a screenshot of the client device that is captured by the *ScreenShooter* class.

From the server side we have almost the same classes than in the client side, except for the *ScreenShooter*, which is not required by the server.

Clairvoyance does not formally consider security and privacy issues for the information exchange between clients and servers. However, the platform includes some level of information protection because the exchanged data is compressed, and thus only nodes knowing the compression algorithm can access that information. Moreover, all people connected to the MANET become automatically visible for the participants; this allows detecting unauthorized users trying to participate in the meeting.

Security and privacy issues can be relevant or irrelevant depending on both, the meeting type and the sensitiveness of the information being deployed in the shared displays. Provided that Clairvoyance was not conceived to address informal meetings in a specific application area, it was kept open, allowing the developers to add security or privacy mechanisms over the services provided by the platform. Thus, they can decide when to protect the information exchange with the mechanisms they consider appropriate.

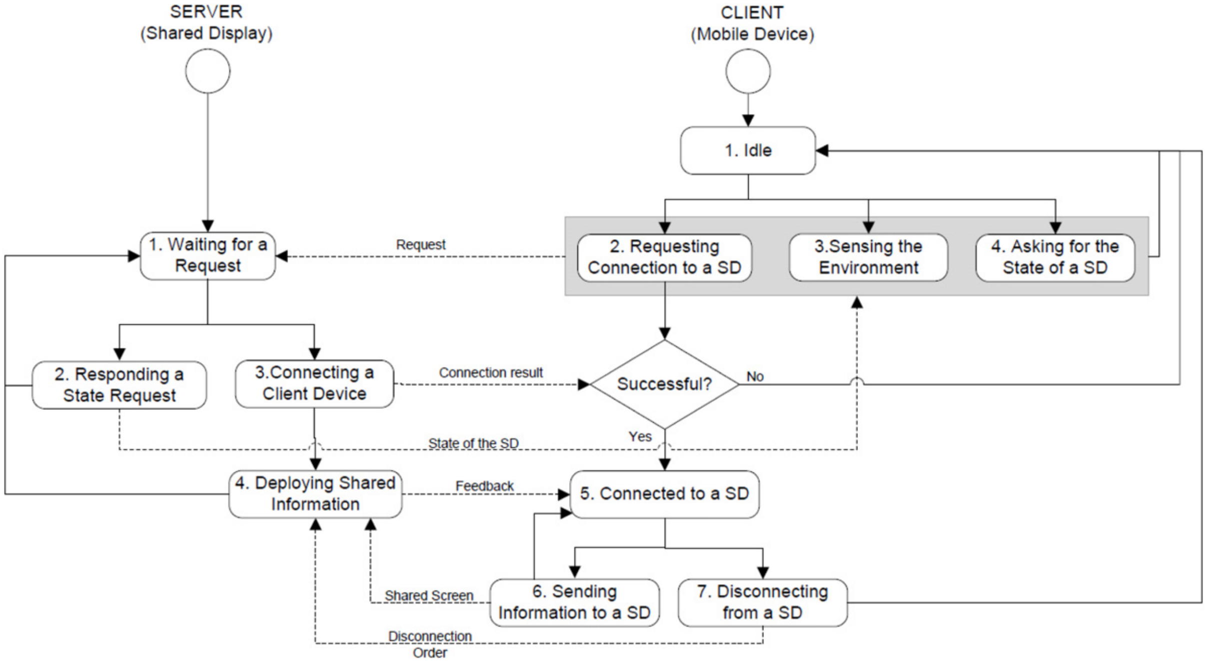

SERVER
(Shared Display)
CLIENT
(Mobile Device)
1. Idle
1. Waiting for a Request
Request
2. Requesting Connection to a SD
3.Sensing the Environment
4. Asking for the State of a SD
2. Responding a State Request
3.Connecting a Client Device
Connection result
Successful?
No
Yes
State of the SD
4. Deploying Shared Information
Feedback
5. Connected to a SD
Shared Screen
6. Sending Information to a SD
7. Disconnecting from a SD
Disconnection Order

Fig. 6. Interaction protocol between client and server.

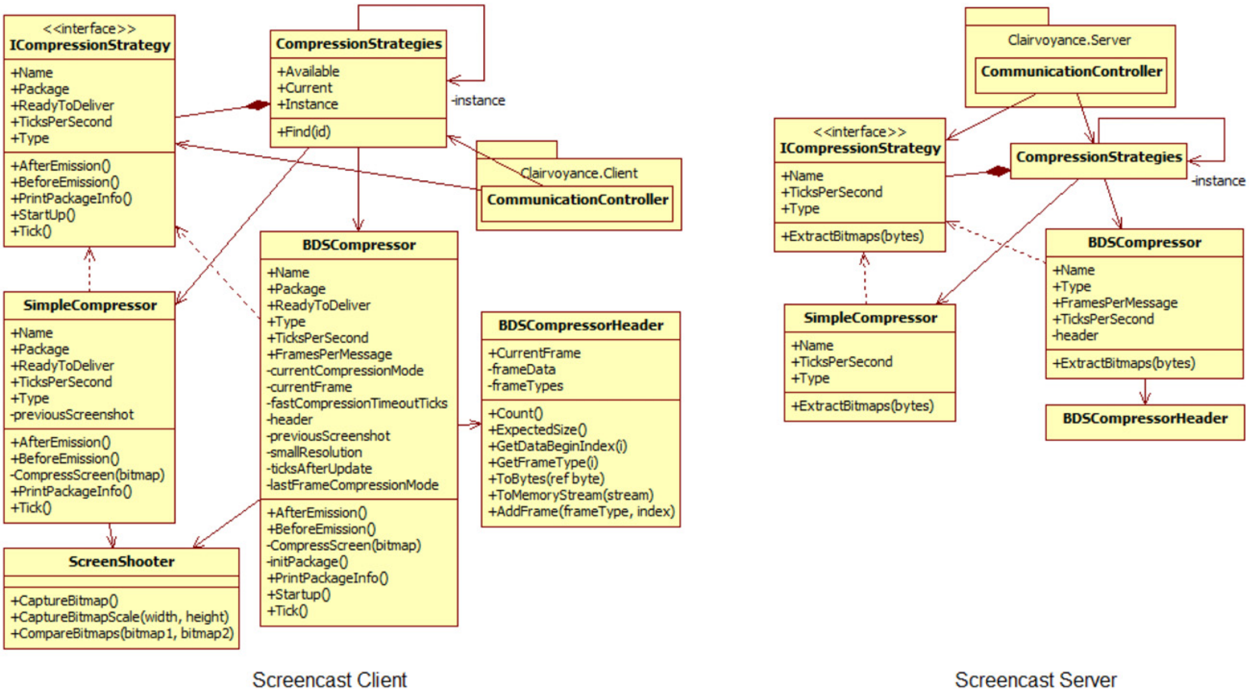
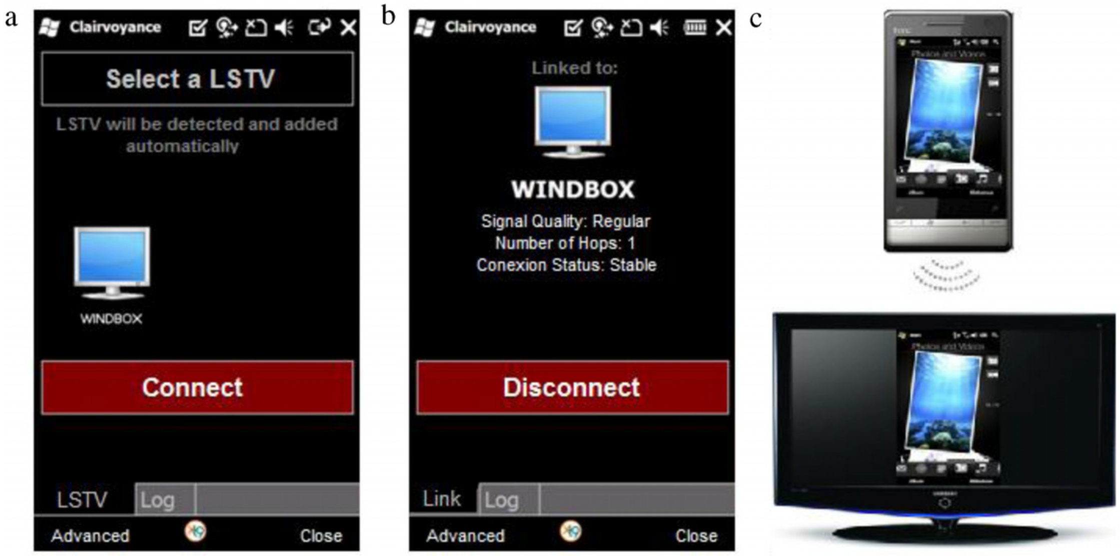


Fig. 7. Structure of the *screencast* component.

3.4. User service interface

This component makes available a list of services through the Clairvoyance API. These services are typically required by developers while building these applications. Table 1 presents a summary of the main services provided by Clairvoyance through this API.

In order to exemplify the use of these services, next we describe a typical process to create a meeting session and present shared information to participants in an informal meeting. When the user of a mobile computing device wants to connect to a LSTV, the Clairvoyance client application needs to become part of a MANET, which is done using the *OpenManet*() service. Once connected to the MANET, the device is able to scan the environment (i.e. the mobile ad hoc network) to get information on all shared displays located within one or more hops of distance (it is done using the *AvailableSD*() service). Fig. 8(a) shows the user interface provided by default by this service, which indicates that there is just one available LSTV that is named WindBox. If the user selects that screen and connects to it (using the *ConnectToSD*() service), the mobile device displays the connection properties (Fig. 8(b)) using the information provided by the *ConnectionStatus*() service. While this link is active, any application or resource displayed on the screen of master



LSTV will be detected and added automatically

Signal Quality: Regular
Number of Hops: 1
Conexion Status: Stable

Fig. 8. Clairvoyance user interface at the client module. (a) Devices discovered in the vicinity, (b) A connection is established and its status displayed on the client screen, (c) The screen of the mobile device is mirrored on the large screen.

Table 1
Summary of Clairvoyance services.

Service name	Description	Client support	Server support
OpenManet( )	This service adds the device to an available MANET, as a client device or a shared display. If a MANET is not available in that environment, this service creates the MANET and adds the node as the only member of the network. Then other nodes can be added to this MANET.	√	√
ManetStatus( )	It retrieves the information of the MANET current structure, including its topology, and node types, availability and distance (in terms of number of hops) for the asking device.	√	√
AvailableSD( )	This function is similar to *ManetStatus*( ) but it informs just the available shared displays.	√	
ConnectToSD( )	It intends to establish a master–slave relationship with a specific shared display. This service requires an explicit user's intervention. If the request is successful, then that resource will be displaying the information of the master device (i.e. the smartphone or laptop) until it disconnects from the shared display.	√	
TryConnectToSD( )	This service is similar to *ConnectToSD*( ), but it does not require user's intervention. The device will automatically try to connect to a specific shared display, and it will inform the user just when a successful connection has been established.	√	
DisconnectToSD( )	It is the counterpart of the *ConnectToSD*( ) service.	√	
DisconnectToSDT( )	This function is similar to the *DisconnectToSD*( ), but it does not require user's intervention. It can be used, e.g., to disconnect a device from a shared display when a timeout has been reached.	√	
ConnectionStatus( )	This function captures and informs various parameters of the current connection between master and slave (i.e. shared display)	√	
SendScreen( )	This service captures and transfers a screenshot of the master device to the shared display at a certain frame rate. That information is automatically deployed on the screen of the slave device.	√	√
FileTransfer( )	This function transfers a file on-demand between two mobile devices participating in a meeting.	√	

device is also shown on the LSTV (Fig. 8(c)). For most resources, e.g. documents and pictures, the user perceives that this deployment is performed in real-time. However, the bandwidth provided by the MANET is not suitable enough to deploy videos. In those cases, the user perceives a very small delay between the video frames (0.2 s approximately), which would be handled using streaming.

All users participating in the meeting are able to scan the environment and detect the presence of the WindBox screen (i.e. the LSTV). If this shared display is busy (i.e. linked to another device), the connection button in the user interface will appear as disabled until the current user frees the LSTV. It can be done through three mechanisms: (1) the user makes an explicit disconnection (using the *DisconnectToSD*( ) service), (2) a timeout for the use of the shared display is reached, or (3) the master device gets disconnected from the slave device. Once the shared display is free, another user can connect to it and share his supporting information with the rest of the participants in the meeting.

All these services are available for developers through the Clairvoyance API, which provides an abstraction layer to manage the network topology, participants in a meeting, awareness information, file transfer and messages passing without the need to deal with the low-level mechanisms typically used in MANETs. This should contribute to ease the development of ubiquitous applications that support informal meetings.

3.5. Implementation issues

The first version of this system was implemented using C# and it was available for Windows Mobile 6.0 (for smartphones) and Windows XP and Windows 7 (for laptops and nettops). The second version of Clairvoyance was implemented in java, as a way to increase the number of operating systems for which this tool was available. Now the system runs also in Android 4.x and Linux.

The weight of the client and server modules for both implementations (i.e. in C# and java) is addressable by the devices which they were designed for. The Clairvoyance client application weighs 1.2 MB and it additionally requires 5 MB of RAM to work properly. In the case of the server module, it weighs 800 kB and requires 5 MB of RAM to support the interactions with the clients.

Concerning the computing power, the system requires a processor speed of 600 MHz or higher. Considering these system requirements we can observe that a large variety of mobile computing devices are currently able to run the Clairvoyance modules. The client and server modules have a setup application that simplifies the installation process.

4. Experimental results

Three evaluations were performed to assess the Clairvoyance usability and usefulness. The first one was done by software developers, and its main goal was to determine the suitability of the framework to support the development of these ubiquitous applications. The second evaluation tried to understand how appropriate are the Clairvoyance services in terms of performance, and how stable are those results. End-users did the third evaluation; its main goal was to determine the usefulness and usability perceived by people. Next sections describe these evaluations and discuss the results obtained.

4.1. Developers' evaluation

Two software engineers participated in this evaluation process. One of them had one and a half years of professional experience, and the other one had 2 years. Both of them received the same introductory talk about Clairvoyance, the documentation about the API, and the requirements to be satisfied by the system that they had to develop. Each engineer developed an ubiquitous application to support social encounters among friends or relatives. The application was simple; it had to allow a user to sense the environment, connect/disconnect devices to a LSTV, exhibit information on a shared display and show the connection status when a mobile device was connected to a LSTV. The system should utilize the user interfaces provided (by default) by the framework, thus avoiding the extra time needed to tune or re-design user interfaces. In summary, they had to develop an application integrating some of the user services provided by Clairvoyance through its API.

The meeting lasted 65 min, and during this period the developers had the chance to ask questions to clarify the requirements or Clairvoyance services. After that, the engineers worked individually and both of them developed their own version of the system. Both applications satisfied all the stated requirements.

The engineers were interviewed after this development process. They estimated that the process took between 15 and 20 h of their time, which was mainly spent to ensure that the services added to the application were right and to provide suitable support to end-users.

Both developers found the Clairvoyance API useful and easy to use, although they had no idea about how the system architecture worked. The only architectural aspect they knew was that the framework adhered to the client–server pattern.

The engineers considered that the Clairvoyance API should be easy to use for an ample variety of developers, because the use of this framework does not require developers with particular skills. Although these results are preliminary, they indicate that the framework would be suitable for developers of these applications.

4.2. Performance evaluation

One of the applications developed in the previous evaluation stage was randomly selected for the performance tests. Thirty instances of the same test were conducted to determine the average performance of the evaluated services. Half of the tests were done using a smartphone (HTC Diamond 2) as client device, and the rest used a laptop (Lenovo Thinkpad X201). The following variables were measured in each test: scanning time, connection time, number of timeouts, channel throughput between client and server, and time to detect a change of status. The *scanning time* indicates the time spent by a Clairvoyance client to sense the environment and determine which devices are present. Likewise, the *connection time* measures the time required by a client device to connect to a LSTV. That period goes from the time that the linking request is sent, until the instant at which the link between the client and server is available for use. The *number of timeouts* indicates the number of connection requests which reached a timeout due to connection problems in spite of the LSTV (i.e. the server) being available. The *channel throughput* indicates the number of bytes per second received by the Clairvoyance server during a connection. Finally, the *time to detect a change of status* indicates the delay between the time the LSTV changes its status (e.g. from busy to available) and the time the rest of the devices detect the new status. Table 2 shows the results obtained.

Table 2
Performance evaluation results.

Variable	Smartphone		Laptop	
	Avg.	Std. dev.	Avg.	Std. dev.
Scanning time (s)	4	2	3	1
Connection time (s)	8	3	6	1
Number of timeouts	0	–	0	–
Channel throughput (kbps)	143	35	228	42
Change of status detection delay (s)	8	3	5	2

As expected, the interaction between the LSTV and powerful client devices (i.e. a laptop) has a better performance than when a smartphone is the client. The throughput of the communication channel is limited by the HLMP API implementation. Although the throughput values could seem relatively low, they are comparable to those obtained by well-known MANET implementation infrastructures, e.g. in Optimized Link State Routing (OLSR) [33].

We can evaluate this throughput in terms of the requirements of the healthcare scenario. For instance, we can try to understand how this throughput affects the time required to deploy X-ray images in LSTVs, as shown in Section 3. A typical high resolution chest X-ray image, which is quite large compared to regular X-ray images, weighs 20 MB approximately. Those images can be compressed using lossless JPEG compression to about 8 MB [36]. Therefore the process to transfer those images from a smartphone to a LSTV requires 56 s, which seems to be an acceptable time for people having an informal meeting. Moreover, if we consider that uncompressed computer tomographies weigh 15 MB approximately, a magnetic resonance weighs about 6.3 MB and an ultrasound image occupies 1.5 MB [37], the transfer time for any of these images becomes shorter. These numbers allow us to expect that the proposed system can be used to share several visual resources typically used in healthcare activities.

4.3. End-users' evaluation

We performed a usability evaluation with thirteen end-users in order to determine whether this performance was acceptable for them. Three of them were between 10 and 15 years old, four between 18 and 25, three between 42 and 45, and three between 56 and 60 years old. People received a basic instruction (2 or 3 min), and then they used the application to complete a series of tasks consisting of 3 connections, 3 disconnections, 2 scans and the deployment of 5 images and one video on the LSTV.

A random device (between smartphone and laptop) was assigned to each user. Five of them used a laptop (Lenovo Thinkpad X201) and seven people used a smartphone (HTC Diamond 2). All participants were able to complete the tasks, and the use of a particular device was not relevant. In order to validate the potential impact that the use of a device has over the users' performance (i.e. the time needed to complete the activities), we compared the means considering a significance of 95%. The confidence intervals for both series (i.e. performance using a laptop and a smartphone) are overlapped ($CI_{95\%}$ laptop: 6.76–18.41 min, and $CI_{95\%}$ smartphone: 6.76–18.07 min), therefore it is highly improbable that there is a relationship between the device being used and the users' performance.

We also explored the relationship between participants' age and their performance. In order to do that we split the participants in four groups according to their ages: 10–15, 18–25, 42–45 and 56–60 years old. Table 3 presents a summary of these results. We did not discriminate the user's performance according to the device being used, because it was already proved that these variables were independent.

Analyzing the confidence intervals (CI95%) for the four groups we can infer there is no statistically significant difference between G1 and G2, and also between G3 and G4, since those intervals have an important overlap. However, we can separate the participants in two groups: young (10–25 years old) and adults (42–60 years old) and analyze the relationship between those participants and their performance. In order to do that we have defined the following null hypothesis (H0): *there is no difference between the performance of young and adult people when they use Clairvoyance to support informal meetings*. The Student's test was used to validate this hypothesis; the significance p was equal to 0.000004, refuting the null hypothesis. Therefore we can expect that young people have a better performance than adults.

After using the system, the participants completed a survey using a 5-point scale to rate each item: good (5), acceptable (4), neutral (3), deficient (2), and unacceptable (1). Table 4 shows the results obtained, where we can see that the significance for every item (when using both laptop and smartphone) is over 0.05. This indicates there is no statistically significant difference in the usability of the system, either with a laptop or with a smartphone. These results are aligned to those obtained for the users' performance.

After this evaluation seven participants asked us for a copy of the software to use it in their own homes. This also indicates that the users felt comfortable with the application.

Since various participants were relatives or friends, after the individual evaluation we organized two meetings, with two different groups. The meetings' goal was to share pictures of some travel the participants had done, but not necessarily together. For these meetings every participant had to bring up to 15 pictures in a mobile device we provided them.

The first meeting included three people and the second one had four persons. Both participant sets were disjoint. The participants' age ranged from 10 to 45 years old; i.e. they belonged to the first three users' categories shown in Table 3.

The meetings had a free dynamics, therefore the participants self-organized their participation during each session. They had to connect their devices to a LSTV, present and explain their pictures, and finally disconnect from the shared display. Both meetings lasted one hour approximately and there was a witness that followed the session dynamics. After the meetings two questions were asked to the participants, and they answered using the same scale than the previous survey. Table 5 shows the obtained results, where we can observe that the usefulness of the system is not affected by the type of device used by the participants.

A short interview with the participants after these meetings allowed us to realize that they found more value in the application once they used it during a real informal meeting. Although these results are still preliminary, they are consistent with those obtained in the previous experiment.

5. Conclusions and future work

Shared displays have shown to be useful to support informal meetings in various scenarios, such as at home, hospitals or business settings. Every day it becomes increasingly important to smoothly integrate the mobile devices used by the meeting participants and large displays available in the physical environment. This integration could allow participants to share and analyze the supporting information in a fast or more effective way.

Table 3
Summary of the users' performance.

Users age	Laptop and smartphone		$CI_{95\%}$	
	# of participants	Avg. time (min)	Lower limit (min)	Upper limit (min)
G1: 10–15 years old	3	7.3	5.97	9.23
G2: 18–25 years old	4	8.4	7.45	8.9
G3: 42–45 years old	3	17.2	8.44	23.3
G4: 56–60 years old	3	16.1	11.62	28.05

This article presents a framework named Clairvoyance, which provides a smooth integration between shared displays (particularly large-screen TVs and computer screens) and mobile devices (particularly smartphones and laptops). The main goal of this framework is to help people to share visual resources during informal meetings or social encounters, avoiding manipulating cables to connect devices or perform device configuration processes. The solution uses a wireless link between a shared display and the mobile device that deploys the visual information, and that integration is transparent for the participants in a meeting. No infrastructure-based communication networks are required in the environment where this solution is used, since Clairvoyance automatically creates and manages the communication links required to perform the operations.

This framework provides an API with several user services that can be utilized by software developers to create solutions that support informal meetings in particular scenarios. These services were used by two software engineers to develop a ubiquitous application supporting social encounters of friends or relatives. Although this evaluation is not enough to get final conclusions, the preliminary results indicate that the Clairvoyance services are reusable, and also useful and easy to use for developers of these solutions.

The performance of Clairvoyance was evaluated in particular settings, as a way to assess how reliable and stable its services are. The results allowed us to identify some improvement areas; however, the services performance and reliability are good enough to support informal meetings.

Finally the application was evaluated by end-users in two instances. The first one involved a list of activities that the users should do individually using the system. The obtained results were encouraging with all participants being able to complete the activities. Particularly young people were highly enthusiastic of using the application and they had the best users' performance.

The second evaluation was done by two groups of people performing a real informal meeting; particularly a social encounter to share travel pictures. In that process the participants used the application as in a real informal meeting; therefore they utilized most of the Clairvoyance services and also decided when to use them. The results obtained showed that users were more enthusiastic with the system than in the previous evaluation instance. Since the user interface of the Clairvoyance services is simple, an ample range of people were able to use it successfully.

Concerning the Clairvoyance limitations we can say that the services performance probably should be improved to handle more properly the transfer of videos or large images (e.g. weighing more than 20 MB) during formal meetings. The current system implementation allows transfer of large-size resources in a LSTV, but this sharing process involves a short delay due to the data transfer performed from the client to the server device. Although these delays were accepted by participants of informal meetings, they could affect the dynamics and time use when the meetings are formal. The evaluation of Clairvoyance as support for formal meetings is part of the future work considered in this initiative.

Table 4
Usability and usefulness evaluation results.

Evaluation item	Smartphone		Laptop		Signif.
	Avg.	Std. dev.	Avg.	Std. dev.	*p*
How suitable was the *response time* shown by the application	3.9	0.69	4.6	0.55	0.08
How *understandable* were the user interfaces of the system	4.3	0.76	4.4	0.55	0.44
How *useful* were the system services to perform the required actions	4.6	0.53	3.8	0.45	0.63
How *reliable* were the services provided by the system	4.7	0.49	4.6	0.55	0.61

Table 5
Usefulness to support informal meetings.

Evaluation item	Smartphone		Laptop		Signif.
	Avg.	Std. dev.	Avg.	Std. dev.	*p*
How *suitable* is the application to support these meetings	4.5	0.58	5	0	0.20
How *useful* could be this application to support informal meetings and social encounters	4.8	0.50	4.7	0.58	0.84

Acknowledgments

This work has been partially supported by Fondecyt (Chile), grant N° 1120207; LACCIR, grant N° R1210LAC002; European Commission, FP7-PEOPLE-2012-IRSES, UBIHEALTH Project, grant N° 316337; and CITIC-UCR (Costa Rica), grant N° 834-B2-228.

References

[1] M. Weiser, Some computer science issues in ubiquitous computing, Communications of the ACM 36 (7) (1993) 75–84.

[2] C.A. Bolstad, M.R. Endsley, Shared mental models and shared displays: an empirical evaluation of team performance, in: Proceedings of the 43rd Annual Meeting of the Human Factors and Ergonomics Society, Houston, TX, USA, 1999.

[3] L. Douglas, D. Aleva, P. Havig, Shared displays: an overview of perceptual and cognitive issues, in: Proceedings of the 12th International Command and Control Research and Technology Symposium: Adapting C2 to the 21st Century, Cognitive and Social Issues, 2007.

[4] A. Genest, C. Gutwin, Evaluating the effectiveness of height visualizations for improving gestural communication at distributed tables, in: Proc. of the ACM Conference on Computer Supported Cooperative Work, Seattle, Washington, USA, 2012, pp. 519–528.

[5] B.E. Plimmer, Using shared displays to support group design; a study of the use of informal user interface designs when learning to program, Ph.D. Thesis, Computer Science Department, University of Waikato, 2004.

[6] S. Greenberg, M. Boyle, J. LaBerge, PDAs and shared public displays: making personal information public, and public information personal, Personal Technologies 3 (1) (1999) 54–64.

[7] C. Gutwin, S. Greenberg, The mechanics of collaboration: developing low cost usability evaluation methods for shared workspaces, in: Proceedings of the IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE'00, 2000, pp. 98–103.

[8] R.E. Kraut, A. Lynn, Streeter. Coordination in software development, Communications of the ACM 38 (3) (1995) 69–81.

[9] A.D. Mejia, J. Favela, A.L. Moran, Understanding and supporting lightweight communication in hospital work, IEEE Transactions on Information Technology in Biomedicine 14 (1) (2010) 140–146.

[10] J.J. Contreras-Castillo, J. Favela, C. Perez-Fragoso, E. Santamaria-del-Angel, Informal interactions and their implications for online courses, Computers and Education 42 (2) (2004) 149–168.

[11] C. Gutwin, S. Greenberg, R. Blum, J. Dych, Supporting informal collaboration in shared workspace groupware, in: HCI Technical Report, Department of Computer Science, University of Saskatchewan, 2005.

[12] C. Berkhoff, S.F. Ochoa, J.A. Pino, J. Favela, J. Oliveira, L.A. Guerrero, Implementing shared displays: a tool for smooth integration of large-screen TVs and mobile devices, in: Proceedings of the 4th International Workshop on Ambient Assisted Living, IWAAL 2012, LNCS, vol. 7657, Vitoria-Gasteiz, Spain, 2012, pp. 278–286.

[13] J. Scanlon, If Walls Could Talk, Streets Might Join In, New York Times, September 18, 2003.

[14] E. Churchill, L. Nelson, L. Denoue, Multimedia Fliers: Information Sharing With Digital Community Bulletin Boards, Kluwer Acad. Pub., Palo Alto, USA, 2003, pp. 97–117.

[15] J. McCarthy, Using public displays to create conversation opportunities, in: Proc. of the ACM 2002 Conf. on Computer Supported Cooperative Work, CSCW'02, New Orleans, 2002.

[16] D.M. Russell, C. Drews, A. Sue, Social Aspects of Using Large Public Interactive Displays for Collaboration, in: LNCS, vol. 2498, Springer-Verlag, 2002, pp. 229–236.

[17] A. Gomez-Goiri, E. Castillejo, P. Orduña, X. Laiseca, D. Lopez-de-Ipiña, S. Fínez, Easing the mobility of disabled people in supermarkets using a distributed solution, in: Proceedings of III International Workshop on Ambient Assisted Living, IWAAL'11, in: LNCS, vol. 6693, 2011, pp. 41–48.

[18] S. Jeon, S. Hwang, G.J. Kim, M. Billinghurst, Interaction techniques in large display environments using hand-held devices, in: Proc. of the ACM VRST'06, Cyprus, 2006.

[19] T. Richardson, Q. Stafford-Fraser, K.R. Wood, A. Hopper, Virtual network computing, IEEE Internet Computing 2 (1) (1998) 33–38.

[20] Y. Bai, Y. Ju, MOVE: a mobile personalized virtual computing environment, Future Generation Computer Systems 28 (6) (2012) 890–899.

[21] B.A. Myers, C.H. Peck, J. Nichols, D. Kong, R. Miller, Interacting at a distance using semantic snarfing, in: Proceedings of the 3rd International Conference on Ubiquitous Computing, UbiComp'01, Springer-Verlag, London, UK, 2001, pp. 305–314.

[22] J. Rekimoto, M. Saitoh, Augmented surfaces: a spatially continuous work space for hybrid computing environments, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'99, ACM Press, New York, USA, 1999, pp. 378–385.

[23] K. Ponto, K. Doerr, T. Wypych, J. Kooker, F. Kuester, CGLXTouch: a multi-user multi-touch approach for ultra-high-resolution collaborative workspaces, Future Generation Computer Systems 27 (6) (2011) 649–656.

[24] R.J.K. Jacob, The use of eye movements in human–computer interaction techniques: what you look at is what you get, ACM Transactions on Information Systems 9 (2) (1991) 152–169.

[25] J. Turner, A. Bulling, H. Gellersen, Combining gaze with manual interaction to extend physical reach, in: Proceedings of the 1st International Workshop on Pervasive Eye Tracking & Mobile Eye-based Interaction, PETMEI'11, ACM Press, New York, USA, 2011, pp. 33–36.

[26] A. Bragdon, H.-S. Ko, Gesture select: acquiring remote targets on large displays without pointing, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'11, ACM Press, New York, USA, 2011, pp. 187–196.

[27] R. Cornejo, D. Hernández, J. Favela, M. Tentori, S.F. Ochoa, Persuading older adults to socialize and exercise through ambient games, in: Proc. of the Workshop on Wellness and HCI, PervasiveHealth, San Diego California, May 21–24, 2012.

[28] K.-F. Huang, C.-L. Hu, A simple mechanism for seamless media streaming and playing services in a ubiquitous home network, in: Proceedings of the 2011 IEEE International Conference on Consumer Electronics, ICCE, 2011, pp. 323–324.

[29] R. Weinberg, Producing and streaming high resolution digital movies of microscopic subjects, Future Generation Computer Systems 27 (7) (2011) 906–913.

[30] D. López-de-Ipiña, S. Blanco, I. Díaz-de-Sarralde, X. Laiseca, A platform for a more widespread adoption of AAL, in: Proc. of 8th Conference on Smart Homes and Health Telematics (ICOST), in: LNCS, vol. 6159, 2010, pp. 250–253.

[31] O. Barkay, D. Buchholz, D. Codorean, D. DeVetter, B. Stevens, Evaluating intel pro wireless display for enterprise use, IT@Intel White Paper, September, 2013.

[32] C. Winkler, K. Pfeuffer, E. Rukzio, Investigating mid-air pointing interaction for projector phones, in: Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces, ITS'12, ACM Press, New York, USA, 2012, pp. 85–94.

[33] J.F. Rodríguez-Covili, S.F. Ochoa, J.A. Pino, R. Messeguer, E. Medina, D. Royo, A communication infrastructure to ease the development of mobile collaborative applications, Journal of Network and Computer Applications 34 (6) (2011) 1883–1893.

[34] D. Lopez-de-Ipina, I. Diaz-de-Sarralde, J. Garcia-Zubia, An ambient assisted living platform integrating RFID data-on-tag care annotations and twitter, Journal of Universal Computer Science 16 (12) (2010) 1521–1538.

[35] J. Rodríguez-Covili, S.F. Ochoa, J.A. Pino, V. Herskovic, J. Favela, D. Mejía, A.L. Morán, Towards a reference architecture for the design of mobile shared workspaces, Future Generation Computer Systems 27 (1) (2011) 109–118.

[36] G. Shaw, A clinician's guide to x-ray systems, Journal of the Royal Society of Medicine 94 (8) (2001) 391–395.

[37] C. Ruggiero, Teleradiology, in: O. Ferrer-Roca, M. Sosa-Iudicissa (Eds.), Handbook of Telemedicine, IOS Press, Amsterdam, The Netherlands, 1998, pp. 65–70.

Christian Berkhoff received a Bachelor in Computer Science from the University of Chile in 2010. His research interests include computer supported collaborative work, mobile networks and applications and human–computer interaction. All his work is enhanced by his experience as an IT consultant for many projects at the industrial production processes, and the financial and educational sectors.

Sergio F. Ochoa is an associate professor of Computer Science at the University of Chile. He received his Ph.D. in Computer Science from the Catholic University of Chile. His research interests include computer-supported collaborative work, software engineering, mobile and ubiquitous computing, and educational technology. Dr. Ochoa is a member of IEEE, ACM and the Chilean Computer Society and sits on the Steering Committee of the LACCIR (Latin American and Caribbean Collaborative ITC Research Initiative).

José A. Pino is a full professor of Computer Science at the Universidad de Chile. His research interests include computer supported collaborative work, human–computer interaction, and software industry studies. He has served as President of the Chilean Computer Science Society (SCCC) and President of CLEI (the Latin American Association of Universities Concerning Information Technology). He has co-authored six books and published research papers in international conferences and journals, including Journal of the ACM, Communications of the ACM, Decision Support Systems, Interacting with Computers, and Information Technology and People.

Jesus Favela is a professor of Computer Science at CICESE, Mexico, where he leads the Mobile and Ubiquitous Healthcare Laboratory. His research interests include ubiquitous computing, medical informatics and human–computer interaction. He holds a B.Sc. from the Universidad Nacional Autónoma de Mexico (UNAM) and an M.Sc. and Ph.D. from the Massachusetts Institute of Technology (MIT). He is a member of the ACM and of the Sociedad Mexicana de Ciencia de la Computacion, of which he is a former president (SMCC).

Jonice Oliveira is a professor of Computer Science at Federal University of Rio de Janeiro (UFRJ), Brazil. Her research interests include knowledge management, social network analysis, mobile computing and computer-supported collaborative work. She holds a DSc, M.Sc. and B.Sc. from UFRJ.

Luis A. Guerrero is a professor of Computer Science at Universidad de Costa Rica. His research interests include computer-supported collaborative work, mobile and ubiquitous computing and human–computer interaction. He received a B.Sc. from Universidad de Costa Rica, an M.Sc. from Instituto Tecnológico de Costa Rica and a Ph.D. in Computer Science from Catholic University of Chile. He coordinates the Ph.D. Program at the School of Computer Science and Informatics from Universidad de Costa Rica.