Extending Instant Messaging to Support Spontaneous Interactions in Ad-hoc Networks

Jesus Favela[1], Christian Navarro[1,2], and Marcela Rodriguez[1,2]
[1]Departamento de Ciencias de la Computación, CICESE, Ensenada, México,
[2]Facultad de Ingeniería, Universidad Autónoma de Baja California, México
{favela, cnavarro, marcerod}@cicese.mx

ABSTRACT
The proliferation of different computing devices such as handhelds and wall-size whiteboards, as well as Internet-based distributed information systems are creating ubiquitous computing environments that provide constant access to information regardless of the user's location. Handheld computers are being transformed from personal electronic agendas into mobile communication devices with intermittent network connectivity. Thus, these devices are becoming a natural medium to tap into an ubiquitous computing infrastructure. Not only do they store much of the user's personal information (contacts list, meeting schedule, to-do list, etc.), but they are always at hand, in sharp contrast with desktop computers. Handhelds, however, most often operate disconnected from the network reducing the opportunities for computer-mediated collaboration with other peers or computational resources. In this paper we propose the extension of an Instant Messaging and Presence Awareness (IM&PA) application running on a handheld device to interact with devices that come within reach of the user as he moves within a pervasive computing environment. Agents are used to represent the devices and appear as first class objects in the IM&PA system, which provides a consistent interface for the user to be aware of the status, and be able to interact with, both people and devices.

Keywords
Ubiquitous computing, opportunistic interaction, software agents, service discovery.

INTRODUCTION
Recent research efforts in CSCW have focused on supporting informal or casual interactions. Among these, instant messaging and presence awareness (IM&PA) systems such as ICQ or Jabber have become surprisingly popular by providing simple mechanisms to let users instantly know the status of other colleagues and engage in lightweight communication with them, either synchronously or asynchronously. Additionally, the recent availability of such services on handhelds and cellular phones is very appealing as evidenced by the success of DoCoMo's i-mode in Japan [5].

The number of users of handheld computers around the world grows at an impressive rate. Personal Digital Assistants (PDAs), smart cell phones and similar electronic devices are now part of our daily lives. As new services and more powerful devices reach the market this tendency will certainly continue in the near future.

Due to the personal nature of handheld computers, most of its applications today are single user oriented and require limited or no connectivity at all. However, the users of these devices work in collaborative environments in which the need to exchange information and share ideas with others is very clear. Additionally, the information stored in handheld computers: dates of meeting and events, contacts information, to-do lists, and e-mail messages, plays a central role in a large number of collaborative applications. Thus, as these devices become more widespread and support network connectivity, we expect them to become major players in future ubiquitous collaborative environments, acting as mediators of the user's personal space and the public infrastructure.

Ubiquitous computing environments provide natural interfaces to interact with a variety of devices (handhelds, laptops, electronic whiteboards, etc.) which provide almost constant access to information and processing resources to mobile users [1]. Furthermore, the interconnected infrastructure as a whole should be able to sense the context in which a specific situation is taking place and adapt to it according to its location of use, the people and objects that are around, and changes of those entities over time. These rich environments provide new opportunities for interaction with colleagues and services offered by devices that are within reach of the user.

A key technical issue faced by the developers of these pervasive computing environments is the discovery of services as users move within an environment full of computing devices. In recent years several technologies have been proposed to deal with the discovery and advertisement of services [7], among the most popular ones being JINI and Bluetooth. However, this work has focused on issues related to the software infrastructure, and to our knowledge no work to this date has dealt with HCI issues. Among them, how will the user be notified of the presence and status of new devices offering services and how will he interact with them, particularly when some of these devices might be new to him.

In the rest of the paper we propose an infrastructure for opportunistic interaction with services available in

pervasive computing environments, named DoMo. DoMo is based on an instant messaging and presence awareness application running on a handheld device. As users join an ad-hoc network, agents representing services are registered in the roster of the instant messaging application. The user then becomes aware of the presence and state of the devices offering services in this network. To communicate with these devices the agent uploads a protocol in the device which is used to create an interface with which the user will communicate with the agent and through it, with the device. Feedback to the user is provided as changes in the state of the device by way of the instant messaging application.

The rest of the paper is organized as follows: Section 2 presents the AIDA handheld IM&PA application upon which DoMo was build. In Section 3 we introduce the architecture of DoMo. In Section 4 we describe the structure of the agents that represent devices and present an example of how it is specialized to support a particular device. Finally, in Section 5 we present our conclusions.

AIDA: IM&PA OF PEOPLE AND SHARED RESOURCES

AIDA is an instant messaging and presence awareness client for handheld devices running PalmOS. It uses a Jabber server (http://www.jabber.org) to obtain status information from other users. The information in the handheld is synchronized with the server every time the device is connected to a point of access, using either IrDA or a serial connection.

As seen in Figure 3a, the interface of AIDA displays the presence of other users. To the right of each user's name, a label is displayed with the connection status of users in the list. Clicking on one of this buttons displays a dialog box that can be used to send messages to the corresponding user. User status in the application can be one of the following: On Line, Disconnected, Away, Busy, or Around, this last one, when the user is in transit from one place to another using his handheld computer, and may establish a network connection shortly.

In addition to acting as an instant messaging and presence awareness application with contacts explicitly added by the user to his roster, AIDA will notify of the presence other users that are classified as available in public facilities. For instance, when a user is attending a conference he might advertise himself to any other conference attendee that wants to contact him, or only to those that meet certain criteria. When another conference attendee synchronizes his AIDA client he will be notified of their availability even tough he didn't added them explicitly to his roster.

AIDA also implements the Doc2U API [6]. Doc2U extends the concept of presence awareness associated to users in instant messaging applications, to shared resources. It supports coordination of collaborative writing activities by introducing the concept of document presence, to provide users with general awareness information of shared Web resources based on an instant messaging paradigm.

Doc2U uses a Web server with WebDAV extensions [8] as a shared repository and the Jabber instant messaging server for notification. Doc2U clients extend the functionality of Jabber by including in the user's roster not only people, but also other resources stored in the web repository. These documents have first class presence in the system. They appear in the client as separate entities with their status indicated with names or icons. Figure 3a, for instance, shows the state of document index2.html as being *Locked*. In AIDA users and resources are differentiated with the capital letters P (people) or D (documents or other resources). Besides allowing interested users to be aware of the state of these resources, Doc2U allows users to interact with these documents (the equivalent of sending messages to another user) to, for instance, put a lock on a resource, or launch a collaborative writing application to edit the document.

By extending the notion of user presence to documents and other resources, AIDA offers new opportunities for casual encounters in a community of co-authors. For instance, when a user notices that a document has been locked, she might find this as an appropriate opportunity to send a relevant message or even join his colleague in a synchronous collaborative authoring session. The Doc2U client can also be used to send subtle messages to grab the attention of his co-authors to the fact, for instance, that he has just finished reviewing the document. The significance of documents and the progress that has been done or remains to be done with them is often communicated in the manner in which these objects are presented or delivered to other people [4].

This extended notion of resource presence can be applied to ad-hoc devices and services. For instance, as a user joins a private network, for instance in an electronic meeting room, new resources might appear as entities of his handheld AIDA client representing for instance a local printer, a projector, or the air conditioning system. Through this application the user can become aware of the status of these devices (on-line, busy, etc.) or initiate interaction with them (print a file, lock a resource, increase the temperature, etc.). The interfaces through which the user can interact with these services can be instances of the more general free-form messages exchanged between users of the instant messaging application, providing a consistent interface for users to interact with other people, documents, devices, and the services they offer.

THE DOMO AGENT-BASED PERVASIVE COMPUTING ARCHITECTURE

We have extended the architecture and functionality of AIDA to support a pervasive computing environment named DoMo. Figure 1 shows the architecture of DoMo, where the following components can be distinguished:

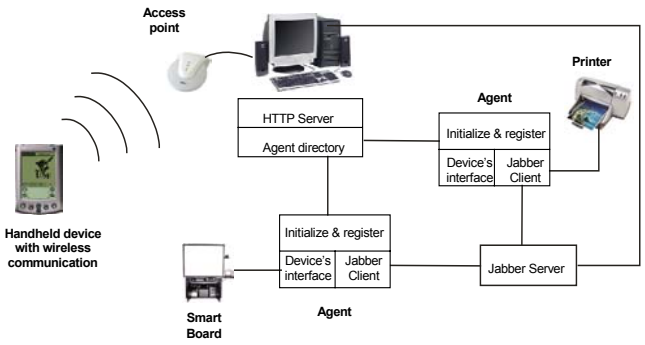


Figure 1. Architecture of the DoMo Pervasive Computing Environment

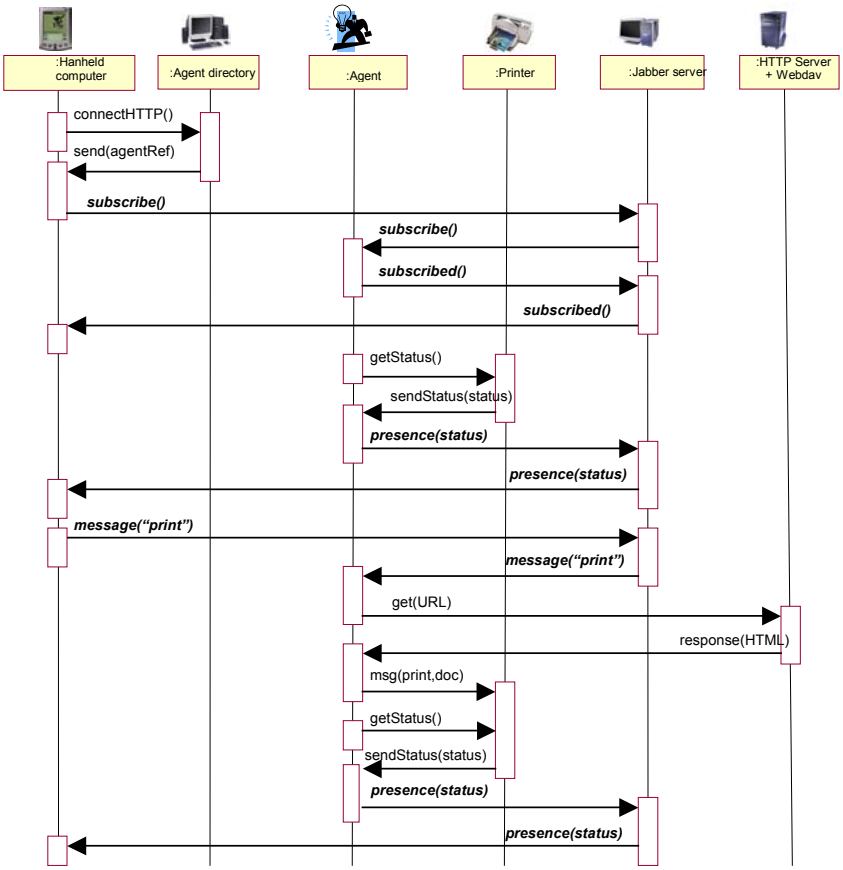
A PalmOS handheld device running AIDA. This is the mobile device used while the user is away from his desktop. The application is also used to collaborate with people carrying similar devices, or with the surrounding environment.

A terminal or desktop computer that acts as an access point. This computer is used to access both networked information and the handheld device. Wireless connectivity to the access point is increasingly a viable option with the proliferation of devices that support the Bluetooth and IEEE 802.11b standards. The access point runs an HTTP server to handle XML/HTML communication with the handheld. In addition, it stores the Agent Directory to which agents representing services will register.

Agents that represent devices. These agents might run in the access point or any other computer with connectivity to the access point. An agent implements a protocol to register into the Agent Directory, an interface to the device that acts as a wrapper to query its status and interact with it, and a Jabber client to notify the device's status to all interested users and receive XML messages through which remote users will interact with the device.

Devices. These are devices that offer services and are connected to the local network. Communication with the device is made through its agent. Devices define possible states, the services it offers, and the protocol used to interact with them.

Jabber server. This server is used to notify the state of people and agents, and to handle the interaction between people and devices through XML messages. All communication between the handheld and the agent will go through this server.

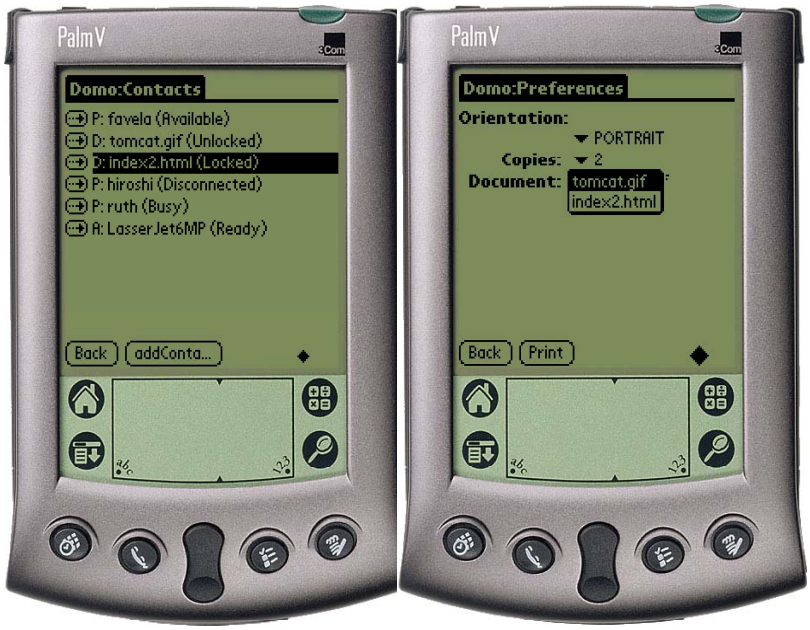


To illustrate how this components interact, we show, in Figure 2, a sequence diagram of a sample interaction in which a user whishes to use a local printer to make a hardcopy of a document stored in an HTTP server. When the user's handheld comes within reach of an access point with an agent directory, it gets in contact with it and

Figure 2. Sequence diagram of a sample interaction with a printer in DoMo

receives a list of agents that have previously registered with it. When AIDA receives the list of agents, it requests a subscription to them through the Jabber server without the user's intervention. The agent might deny the subscription, as often users do, due to privacy or security reasons. For instance, the agent configuration might restrict the use of the printer only to a predefined list of users.

Once the Jabber server notifies AIDA that the subscription request were accepted, the agents representing the devices, a printer in this case, will be included in the roster and their name and status will be displayed. The user can know select the printer's agent from the roster. Doing so will display a form in which the user can select the document he whishes to print and any other printing options he whishes to set, such as the number of copies to be printed. This information will be sent as an XML message to the agent via the Jabber server. The agent will request the document to the web server and use its interface to the device to print the document as requested. While the document is being printed, the state of the device will change, this will be notified by the Jabber client included in the agent to all users that have registered the presence of the device. This is how the user who requested the document to be printed gets feedback from the operation just performed. Notice that in the diagram shown in Figure 2 all events in bold correspond to the Jabber's protocol.



a) b)

Figure 3. Using AIDA to print a document in the DoMo environment

AGENTS AS PROXIES FOR DEVICES AND SERVICES

Agents in the DoMo infrastructure are software components that run as daemons on computing devices with connectivity to an agent directory and a Jabber server. Agents provide a standard mechanism to initialize and register the agent with one or more agent directories. During initialization, the agent confirms that the device that it represents is reachable and obtains its current status. The first time the agent is run it will register the device in the Jabber server as well.

The agent also includes a Jabber client, which is used to notify changes in the state of the device and receive messages from users. All commands sent to the agent are received through the Jabber client. Communication from agents to users is achieved by notifying of state changes in the device.

Additionally, the developer that wants to include a new device in the DoMo environment will need to implement an interface to the device within the agent. These are methods that query the state of the device and can instruct it to perform the services requested by the user. The states of a printer, for instance, can be: on-line, off-line, printing, paper-jam, etc. An air conditioning system could have the following status: on, off, current_temperature = N. The printer's interface will also be able to download a document given a URL and send it to print to the device.

An important issue is how the user will communicate with an agent representing a device he has never seen before. Clearly, we don't want to include all kinds of device drivers in a handheld computer with limited storage resources. Even if enough storage is provided this solution will limit the use of the system to devices and services that are known a-priori. We deal with this issue by creating an XML application to specify devices and services and a protocol to interact with them. Thus, the programmer that creates an agent to incorporate a new device in DoMo, needs to write an XML document that is valid with respect to the DTD that we have defined for this purpose. This

XML document indicates the type of device represented by the agent, and it defines the contents of the interface that will be shown to the user by the AIDA application when the agent is selected and through which he will be able to interact with the device. This is illustrated in Figure 3. Figure 3a shows AIDA with an agent for a printer called LaserJet6MP. Figure 3b illustrates the interface presented by the user when the agent is selected. It shows a list of documents that can be printed (those documents to which the user has previously registered and which are displayed in the roster), and the options available for the printing service, namely, the number of copies to be printed and the orientation of the page. Figure 4 shows a segment of the XML document that defines the service represented by the agent.

```
<device>
  <deviceType> Printer </deviceType>
  <deviceName> LaserJet6MP  </deviceName>
  <protocol>
     …
    <choiceGroup>
      <choiceName> documentsChoice </choiceName>
      <title> Documents: </title>
      <Array>
        <arrayName> documentsArray </arrayName>
      </Array>
    </choiceGroup>
  </protocol>
</device>
```

Figure 4. XML document defining the service represented by the Printer agent

AIDA was implemented using Java (J2ME), which has the advantage of running on a variety of devices, other than handhelds with PalmOS. A significant advantage for ubiquitous computing environments. Figure 5 shows AIDA running of a cellular phone that supports J2ME. It illustrates the integration of remote-controlled videocamera. In this case, the state of the device specifies the coordinates of the camera, the user can move the device in the direction he desires, and can download an image from the device. This type of device could support security applications. The camera could change its state when it detects movement and notify the user and download an image for him to see.

However, due to the security restrictions of the J2ME environment, applications are not allowed to access or modify other data in the handheld. To deal with this limitation we plan implement a new version of the system using the COMAL handheld collaborative development framework [2,3]. This way, new data items could be added to the handheld as a result of requesting a remote service. For instance, an agent might use this mechanism to schedule a date for the user, which will be directly added to the Palm's DateBook application.

Figure 5. XML document defining the service represented by the Printer agent

CONCLUSIONS

Handheld devices provide natural interfaces to interact with pervasive computing environments. We propose an extension of the instant messaging and presence awareness paradigm to support the discovery of new devices and services, become aware of their presence and status, and be able to interact with them opportunistically.

An agent architecture is proposed to easily add new devices to this infrastructure. Developers that wish to add a new device to the DoMo pervasive computing environment need only to program an interface to the device and define an XML document to specify the interaction with the services it provides. No changes are required to the AIDA handheld application used to interact with the environment.

REFERENCES

1. Abowd, D., Mynatt, E.: Charting Past, Present, and Future Research in Ubiquitous Computing. ACM TOCHI, Vol. 7, No 1 (2000), 29-58
2. Alba, M., Favela, J.: Supporting Handheld Collaboration through COMAL. Sixth International Workshop on Groupware, CRIWG'2000. (2000) 52-59
3. Favela, J., Alba, M., Rodríguez, M.: Autonomous Agents for Ubiquitous Collaborative Environments. Advances in Artificial Intelligence, LNAI 2313, Springer, (2002) 351-360.
4. Harper, R., Procter, R., Randall, D., Rouncefield, M.: 'Safety in Numbers': Calculation and Document Re-Use in Knowledge Work. In Proc. GROUP'01. ACM Press. Sep. 30-Oct. 3, 2001, pp. 242-251.

5. Mitsuoka, M, Watanabe, S., Kakuta, J., Okuyama, S.: Instant Messaging with Mobile Phones to Support Awareness. In Proc. SAINT2001. San Diego, CA, Jan. 7-12, 2001, pp. 223-230.
6. Moran, L., Favela, J., Martinez, A., Decouchant, D.: Document Presence Notification Services for Collaborative Writing. Proc. of CRIWG'01 (2001) 125-133
7. Richard, G.: Service Advertisement and Discovery. IEEE Internet Computing. Vol 4, No. 5. (2000) 18-26.
8. Whitehead, E.J, Wiggins, M.: WEBDAV: IETF Standard for Collaborative Authoring on the Web. IEEE Internet Computing, Vol. 2, No. 5. 1998, pp. 34-40.