A Pervasive Reminder System for Smart Homes

Sylvain GIROUX and Simon GUERTIN

Département d'informatique,
Université de Sherbrooke
2500 boul. Université, Sherbrooke, Canada J1K 2R1
{Sylvain.Giroux, Simon.Guertin,}@USherbrooke.ca
http://www.dmi.usherb.ca/~sgiroux/domus/

Abstract: One of the research projects at the DOMUS laboratory aims at the realization of smart homes and cognitive assistance for cognitively impaired people. The aimed system is a pervasive information system able to detect failure and anomalies in the behaviour of the habitants, and then to help them and notify other systems or people about potential danger. In this paper, we address the issue of finding the user and choosing the most appropriate device to transmit a message. As a prototype, a pervasive reminder system for message management explores the possibilities of dynamic cooperation between different kinds of devices, hardware, and software communicating through different networks. This pervasive system tries to locate, identify and notify the resident of upcoming appointments. The issues addressed are 1) localization and identification of the resident; 2) finding devices for communication and delivering messages according to the position of devices and the position of the user 3) spontaneous networking, discovery services, and service composition, and 4) migration of a running application. The latest allows a user working on a device to switch to another device more adapted to her task, making the application really pervasive.

1.1 Introduction

As Weiser said about pervasive computing, « The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it. » [1]. The DOMUS laboratory of the University of Sherbrooke studies pervasive computing and mobile computing. One of the research projects aims at the realization of smart homes and cognitive assistance for cognitively impaired people [2] [3]. People suffering from cognitive impairments —Alzheimer disease, schizophrenia, brain injuries...— often have no choice other than living in medical institutions. Smart houses [4] and mobile applications, e.g. activity compass [5], can play a central role in keeping them at home in their community. The aimed smart home would be a pervasive information system able to detect failure and anomalies in the behaviour of the habitants, and then to help them or to notify other systems or people about potential danger.

In this paper, we address the issue of finding the user and using the most appropriate device to transmit a message. As a prototype, a pervasive reminder system for message management explores the possibilities of dynamic cooperation between different kinds of devices, hardware, and software communicating through different networks. This pervasive system tries to locate, reach the user with the nearest or most convenient device and notify her of upcoming appointments. To do that, we need at the implementation level to address spontaneous networking, discovery services, service composition, and migration of a running application

First we sketch the general issues raised by smart homes for cognitive assistance (§2). Then we describe the prototype: the physical set up (§3) and the functional requirements (§4). Once the goal and setting are clear, we detail the distributed implementation of the various components (§5).

2 General Issues

Let suppose Mrs Smith is suffering from the Alzheimer disease. She is cooking a stew and preparing coffee when somebody rings at the front door. She goes to answer, chats a bit, and forgets something is cooking. And she listens TV in the lounge. The smart home detects the situation is hazardous: nobody is in the kitchen and the stove is on for a long time. The system decides to send a reminder to her. So the issue to solve is first to find where she is, then find what devices in the lounge can be used to present the reminder — TV, telephone, her PDA…—, and finally package the message and send it to the chosen device. Hence to finally provide Mrs Smith with the appropriate feedback, a pervasive system must face many issues. In this paper, we explore the combination of the following ones:

- *Identification and localization of the appropriate user*: there may be many people simultaneously in the house; they can move from room to room.
- *Heterogeneity of devices, hardware, software and networks protocols*: in a house, there is a great variety of coexisting devices (electrical appliances, computers, TVs, PDAs…), software (server and client components, different operating systems and programming languages…), and networks (Ethernet, WiFi, Bluetooth, X10 [6]). They all have to collaborate.
- *Spontaneous networking and device and service discovery*: it is not possible to configure and to know beforehand all the devices and systems that will be present in the house; devices may join or leave the house dynamically; some are mobile, other are stationary.
- *Transitory coalition of devices*: devices and systems may offer complementary services that must collaborate for a moment to perform a task, for instance to deliver the message to a given user.
- *Pervasiveness*: interactions between components must be invisible and transparent to the user, furthermore a computation may have to migrate from one device to another. For instance, if the telephone is first used to reach Mrs Smith and she asks to see the state of the stove on TV, using the camera in the kitchen.
- *Zero-configuration*: obviously Mrs Smith cannot be asked to configure the system.

3 Prototype : a Pervasive Agenda System

The physical and software setup of the prototype incorporates multiple technologies that are not directly designed to work together (Fig. 1). In particular, the standard electrical hardware will be interconnecting with the computer hardware and software systems through the X10 protocol. A system may use different electrical devices to enhance the user's level of comfort or to extract information about different physical aspects of the house. X10 devices are affordable and are already made for home automation. The X10 devices used in the project are:

- one X10-computer interface device connects to the serial port of a computer on one end and into an electrical outlet on the other end. The device enables to read and write X10 codes on the electrical wires. Model: X10 TW523 Two-Way Interface Module.
- two wireless motion detectors. Model: Version II Wireless X10 Motion Sensor.
- one RF base to convert the signals from the wireless motion detectors into the electrical system. Model : X10 16 Device RF Base.

To read and write X10 codes from and to the computer interface, we use standard serial port programming in Java. Other devices used are

- a PC running Linux Red Hat 9 mostly used to run different services of the Domus infrastructure.
- the user's PC running windows XP in the main room.
- a standard wired LAN setup
- a wireless transmitter base to allow WiFi connections to the network.
- a standard home router with DHCP capabilities is used to interconnect the networked hardware and allow connections to the Internet. Model: Lynksys Cable/DSL 4-Port Router.
- a printer enhanced with a Bluetooth antenna to allow printing from diverse devices.
- a HP iPAQ Pocket PC h5550 with Bluetooth and WiFi capabilities.

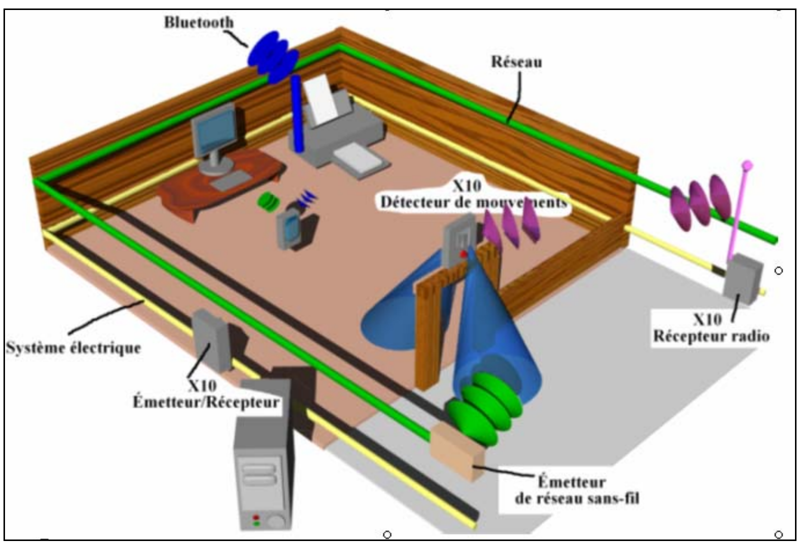

Figure 1 Current prototype implementation of the pervasive agenda system.

4 An Active and Pervasive Agenda System

The prototype built implements an appointment manager or agenda that keeps track of a user's appointments like any other agenda would do. Where the system distinguishes itself is that it is a distributed application that uses other distributed applications offering different services throughout the house. The agenda operates in conjunction with a messaging system to try to inform the user of an upcoming appointment. The system also uses other features from the house to better accommodate the user's needs.

To perform its duties, the system first must search for a device to notify the user (§4.1), then localize the user and prepare the delivery of the message to the user (§4.2), and finally support the migration of a user session from device to device to follow a user from room to room (§4.3). The next sections will present how these issues were addressed in the prototype.

4.1 Search of Devices for Message Delivery

When an appointment is approaching, the system has to notify the user. To do this, the system attempts to use any service implementing the messaging service. The system might or might not know where the user is in the house, so it disseminates the message to any resources that may communicate the information to the user. When the user reads the message, the system gets a confirmation the user got the message. The system then stops trying to reach the user and notifies other messaging services in the house so they can remove the read message.

4.2 Preparing Message Delivery to a Nearby User

When the system knows a certain user is nearby, it prepares to log the user in. Due to limited means of user identification in the current setup, we consider only one user. In this scenario, motion detectors work in collaboration with other software services running in the Domus infrastructure. The agenda is usually in standby mode in the main room. When the user enters the room as detected by motion detectors, the agenda login screen is filled with user name. If the user wants to use the application, she just needs to type in her password and start using the tool. The agenda program can do this because it uses a service that informs any client about the presence of a user in the room. When the user leaves the room, the agenda is also informed and reacts accordingly by removing the pre-filled login.

4.3 Migrating user session from device to device

A user may need to migrate from one device to another when working with an application (in our case in the agenda). She may need better display capabilities. She may prefer to work in another room. She may need to work on a wireless device. When the user switches device, she wants to pick up the session where she left it. There are three cases when this happens:

- the user directly switches to the other device if it is available;
- the user stores her session on the server if no available device meets her needs;
- the user session is retrieved from the server when the device becomes available.

In the first two cases, the agenda system will log off the user from the current device when the transfer is successful. A login screen with the user name already filled in will be displayed to the user on the other device. The session will then be restored to where it was left on the previous device. In the second case, the user is logged off from the current device when the session is transferred to the server successfully. The user will then have to log on the appropriate device when it becomes available (third case). The session can then be restored from the server from the agenda's menu commands (or it could be automatically restored on a successful login).

5 Implementation

In this section we present implementation details of the prototype. First we explained the choices underlying the general infrastructure (§5.1). Next we present general services that must run in any cases (§5.2). Then we present specific services: services related to X10 components (§5.3) and services related to message delivery (§5.4). Finally the agenda system as a whole is described (§5.5).

5.1 Jini and Java as Backbone for the General Infrastructure

Indeed we want to build dynamic distributed system on heterogeneous components and supporting code migration. Hence we choose Java[1] and Jini. Java provided the compulsory multi-platform support and code migration through RMI. Jini provided

- *spontaneous networking*: services offered to the users or to other services may join and leave anytime.
- *service registration and discovery*: Jini discovery services enable to find the needed services in the DOMUS federation.

Reggie[2] is running on the Linux PC in a persistent and activable mode to save CPU resources. An http server is running also on this computer to make the necessary Jini jar files available for download on the house's LAN.

There is no centralized system but many different kinds of services that can have different physical or logical location. Services are grouped in the main federation: DOMUS. In fact, many sub-federations are used to realize the whole system (Fig. 2). A federation could either represent a room, a floor, the whole house or any logical boundary. With a good use of federations, a service or client can better choose a specific service depending where the desired service is registered, giving hints on its location.

5.2 General Services

For the Domus infrastructure, we developed a Domus server. This server is a service that provides basic information about this infrastructure. It also provide another HTTP server that allows to upload files that other services might need to store and make available to other programs that will use the system. Those files contain the code to distributed for a Jini service to work. These files need to be available through an HTTP server. This feature simplifies the development of new services.

The security server is a service that is used by any other servers or client program that needs to authenticate a user. This service also runs on the Linux PC. It is a small database that stores usernames and passwords.

5.3 Services related to X10 Components

The X10 server allows the X10 technology to be incorporated in the Domus infrastructure. This server uses serial port programming in Java to interact with the X10 Computer Interface (§3). This computer interface monitors X10 codes travelling on the electrical

[1] We used JDK1.4 with the latest release of Jini. We needed to downgrade the version of Java we used when we incorporate the iPAQ because J2ME is only compatible with the JDK1.3.
[2] Reggie is the lookup service used in Jini to allow services to register themselves and lookup other services.

wires and stores them in the device's memory. With the X10 server, other services can read or write those codes to know the state of devices or to control them. To use it, a client or service of the Domus federation registers to the X10 server and specifies which codes it is interested in (Fig. 3.).

The room's motion detector and darkness detector service allows other services to show a more adapted behaviour in the presence of a user. This service is available to any other client or service interested in the state of occupation of the room or in the state of lighting. This service is composed of

- the two motion detectors that are placed back to back above the room's entrance;
- the wireless base that transmit the signals of the two wireless detector's X10 into the electric circuit of the house;
- the X10 server;
- the motion detector program that will handle the X10 signals.

The motion detector program registers to the X10 server and specifies X10 codes it needs from the motion detectors. It will then analyze the X10 codes received from the X10 server to determine either the room is empty or not, in full darkness or not. The algorithm can detect leaving or entering the room by analyzing the time elapsed between the X10 codes thrown by each motion detectors. Other services can then register themselves to the motion detector service to be informed of the room's occupation status.

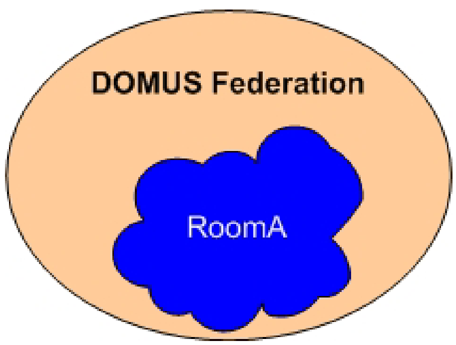

Figure 2 Organizing Domus into sub-federations.

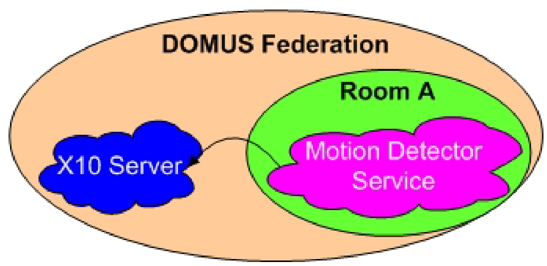

Figure 3 Detecting presence with X10 components.

5.4 Messaging System Server

To deliver messages to users in the house, we implemented a message server. This message server accepts request from clients that wants to deliver a message to a specific recipient. The message server processes the requests and looks up all the message clients in the Domus federation to deliver the message. The server can handle new message clients joining the federation. When a message has been read and confirmed by the recipient, the message client informs the message server of the successful delivery. The message server then proceeds to inform the originator of that message of the successful delivery. The message server also has to inform all the other message clients to remove this specific message from their queue.

Tasks of message clients are simpler. They have to display the message any way they want but they must keep a certain degree of confidentiality and integrity. The message client receives message request from the message server and then waits for a user interaction. For a user to read a message, she normally enters her password for confirmation. The message client confirms the user's identity with the security server. After successful user authentication, the message client then contacts the message server to acknowledge a successful delivery. Message clients must have the basic functions of adding a message and removing a message when requested by the server.

5.5 The Agenda System

The agenda server is a service that handles appointments for the users and tries to notify them of an upcoming appointment. The server has all the basic functionality an agenda server would have. That is to create, store, retrieve, and delete appointments. It uses the security server to authenticate users upon connection of clients. What makes the agenda server unusual is when time comes to notify a user. It searches the Domus federation for messaging services able to send out the notification to the right recipient (Fig. 4). Once the server has received a confirmation of reception by the messenger service it stops and marks the appointment as acknowledged by the user. The server also takes care of overdue appointments that were not acknowledged and the server removes them from the messenger service.

The server also supports the storage and retrieval of a user's session if she wishes to switch device. At the beginning of a session, the user gives a user name and password. She then uses the agenda like a normal agenda to browse, add and delete appointments. If she wants to switch from one computer to another that better suits her needs, she can browse the available computers and directly migrate her session to the new computer if it is available. If no computer is available, the user can store her session to the agenda server to retrieve it later from any other computer when available. When switching directly to another computer, a login screen with the user name is automatically pre-filled. The user enters her password and the session is restored as it was left on the previous computer. (Fig. 5.). There is no need to reopen windows or retype the information, even if not saved.

In the project we assume that it is always the same person that enters or leaves the room. When the client is started, it is looking for the motion detector service to be informed of the presence of a user in the room. There might be multiple motion detector services in the whole system, but the client will only be interested in a motion detection service that belongs to the same Jini federation as the agenda service. This permits the agenda to start the login process when someone enters the room. If there is no motion detector service for the room, the agenda client just behaves normally without inferring somebody is in the room. The client is also informed of the lighting of the room through the motion detector service. It could then open up a light or any other service that would enhance the environment for the user if she chooses to use the agenda (this is not yet implemented).

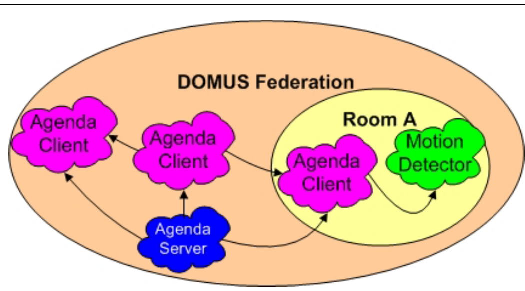

Figure 4 Agenda clients and server.

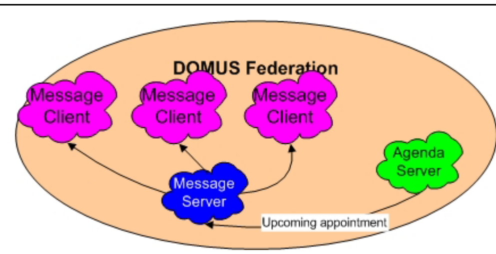

Figure 5 Request from the server to notify a user.

6 Conclusion

Smart homes providing cognitive assistance for cognitive impaired people raise many difficult issues. In the long-term, the DOMUS laboratory wants to implement such a smart home as a pervasive information system able to first detect failure and anomalies in the behaviour of the patients, and then to help them on-site and to perform some sort of tele-monitoring. Because it is a very complex task, we decided to address a simplified version of this problem.

In this paper we thus described the implementation of a pervasive distributed agenda system. This system has been used to explore how sensing and information devices can be integrated using a variety of communications services on the one hand and how information can be delivered to the user in a pervasive environment on the other hand. The emphasis was put on locating and communicating information to residents. The prototype enabled to explore the possibilities of dynamic cooperation between heterogeneous devices, hardware, and software communicating through different networks. This resulting pervasive system tries to locate, identify and notify the resident of upcoming appointments. Among the issues that have been addressed, let stress: 1) localization and identification of the resident; 2) finding devices for communication and delivering messages according to the position of devices and the position of the user 3) spontaneous networking, discovery services, and service composition, and 4) migration of a running application. The latest allows a user working on a device to switch to another device more adapted to her task. The user can then go on with the application without losing her work, making the system really pervasive. This distributed system is adaptive to changes of availability of services, devices and computers.

We intend to use the agenda system as a reminder system in an Agenda Home [7]. Robustness will then be a key issue to address. With respect to monitoring in a smart home, we need a way to store the vital information in case the network or a service is down. When the service or the network would become available again, the services would then report back the information that was saved locally.

Finally we intend to exploit the limited range of Bluetooth to build local coalition of devices based on physical proximity.

7 References

[1] Mark Weiser, *The Computer for the 21st Century*, Scientific American, 1991.

[2] H. Pigot, et al., *The intelligent habitat and everyday life activity support*. *5th international conference on Simulations in Biomedicine*, Slovenia, April 2003.

[3] Pigot H., et al., *The role of intelligent habitats in upholding elders in residence*. *5th Intl Conf. on Simulations in Biomedicine*, 2003, Slovenia. WIT Press

[4] Rialle .V, et al. *An experimental Health Smart Home and its distributed Internet-based Information and Communication System*. *MEDINFO 2001,* IOPress. pp. 1479-1483.

[5] Kautz, H. et al. *An Overview of the Assisted Cognition Project*, AAAI 2002, workshop.

[6] Boone, K, *Using X10 for Home Automation*, http://www.kevinboone.com/x10.html

[7] Pigot, H., and S. Giroux, *Keeping in Touch with Cognitively Impaired People: How Mobile Devices Can Improve Medical and Cognitive Supervision*, 2nd International Conference on Smart Homes and Health Telematics ICOST 2004, Sept 15-17, 2004, Singapore.