An Architecture for Sentient GPRS-enabled MicroBots*

Diego López de Ipiña Juan Ignacio Vázquez Jonathan Ruiz de Garibay David Sainz
Faculty of Engineering (ESIDE) *Deusto Foundation*
Universidad de Deusto
Avda. de las Universidades, 24 48007 Bilbao (SPAIN)
{dipina, ivazquez}@eside.deusto.es *{joruiz, dsainz}@tecnologico.deusto.es*

Abstract **–Ambient Intelligence is a user-centered concept which combines several computing disciplines with the purpose of enhancing/facilitating the user's daily activities. We deem that autonomous or semiautonomous (remotely controlled) Sentient MicroBots may also be first-class citizens within Ambient Intelligence. Those MicroBots would interact with their surrounding environment assisted by their built-in sensors, effectors and communication facilities, on behalf of the users they serve. In essence, they would also profit as users do from Ambient Intelligence to achieve their programmed goals. In this paper, we contribute with a solution to enable the real-time remote control of GPRS-accessible semiautonomous Sentient MicroBots. Moreover, we discuss the extensions necessary to convert those MicroBots into care assistants for disabled and elderly people.**

Index Terms – Ambient Intelligence, Microbotics, Wireless, Middleware, PDA.

I. Introduction

Ambient Intelligence (AmI) [1] involves the convergence of several computing areas: Ubiquitous Computing and Communication, Context-Awareness, and Intelligent User Interfaces. Ubiquitous Computing [2] means integration of microprocessors and computer services into everyday objects like furniture, clothing, toys and so forth. Ubiquitous Communication enables these objects to communicate with each other and the user by means of ad-hoc and wireless networking. Context-Awareness implies adding sentient capabilities both to the environment and the user mobile devices so that they can understand the current context of the user and offer her suitable services. An Intelligent User Interface enables the inhabitants of the AmI environment to interact with it in a natural (voice, gestures) and personalised way (based on preferences and context).

Unfortunately, the Ambient Intelligence concept is still far away from becoming reality. For instance, it assumes that our living/working environments are equipped with all sorts of computation/communication mechanisms, not publicly available or too expensive for most of us.

Robotics is slowly moving from exclusively industrial contexts to more exoteric areas, such as domestic [3] or air space [4] domains. The Mars Rovers, unmanned land vehicles for exploration of the planet Mars, are probably the most popular robots in the world. Those NASA produced robots posses sophisticated communication mechanisms capable, among many other things, of sending us pictures of the surface of Mars. Nevertheless, most conventional robots have very primitive, if any, communication mechanisms. Obviously, the data transmission technology used by the NASA robots is not commonplace. Thus, it would be interesting to correct this deficiency by making use of readily off-the-shelf technology such as M2M (Machine-to-Machine) communication and public GPRS/UMTS networks. The access through GPRS or UMTS gives global accessibility to those robots, despite their still expensive although slowly decreasing data transfer costs.

This paper describes our experiences combining the promising fields of robotics and M2M, in order to achieve real-time telemetry of sentient microbots over a GPRS data network. Those microbots are equipped with sensors to acquire context, effectors to undertake actions and some built-in logic to both notify and receive commands from remote users, and to act autonomously. Those MicroBots may be used as portable ambient intelligent devices that can undertake tasks on their own or offer us services controllable remotely, independently on where they are located.

The structure of this paper is as follows. Section 2 presents the COMMBOTS concept, our proposal to enhance conventional MicroBots with sentient and communication facilities. Section 3, describes the architecture designed to allow real-time control of those Sentient MicroBots through public GPRS data communication networks. Section 4 enumerates some performance results. Section 5 describes CareBot, an extension of COMMBOTS which proposes robots as carers of people with disabilities or advanced age. Finally, section 6 gives some conclusions and suggests further work.

II. Communicating MicroBots

The COMMunicating MicroBOTS (COMMBOTS) project aims to create a group of MicroBots equipped with mobile communication (GPRS), location (GPS), sensorial and reactive facilities. Those robots can be used, for example, in dangerous environments where they can gather data and materials, or for surveillance and alarm control in security zones.

Fig. 1 shows the two main entities defined within COMMBOTS: (1) MicroBots and (2) Control Stations. The MicroBots are devised to carry out diverse tasks, such as: obtaining and measuring data by sensing, collecting samples, or recording and transferring images to Control Stations. Noticeably, a MicroBot is bundled with an owa22A [5] mobile communication module (GSM/GPRS), by means of which it can receive commands from the Control Stations and send responses. This module is equipped with a 60 MIPS CPU ARM7 and a 2 MB RAM, runs a cut-down Linux OS and offers a C API for its programming. It is therefore a powerful platform not only to communicate data but also to undertake internal processing.

* This work was supported by a 2003-04 SAIOTEK grant from the Basque Government and the Cátedra de Telefónica Móviles at the University of Deusto.

The Control Stations in COMMBOTS are available in two forms: (1) fixed stations operated from a PC and (2) mobile ones operated from a PDA or mobile phone. These Control Stations allow a user to monitor and control at *anytime* and *anywhere* the whereabouts and actions of a fleet of MicroBots. Likewise, the MicroBots may notify events of interest to the Control Stations (e.g. temperature too high, movement detected) or delegate the processing of some data captured.

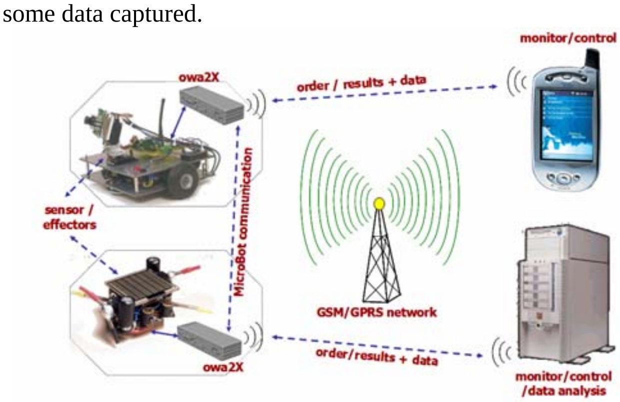

Fig. 1 The COMMBOTS System.

In a nutshell, the main purpose of COMMBOTS is to combine the latest developments in M2M and MicroBotics, to design a GPRS-based real-time remote control system for Sentient MicroBots. Those microbots are equipped with enough sensorial inputs, effectors and processing power to enable their semiautonomous operation, without continuous control from users.

III. The COMMBOTS Architecture

COMMBOTS presents a *client/server/client* architecture devised to ease the end-to-end communication between the Control Stations and the MicroBots. Fig. 2 shows the different components of this architecture together with the data and control flows exchanged among them. In what follows, we detail those components: (1) MicroBots, (2) MicroBots Proxy and (3) Control Stations. Moreover, the MicroBot Protocol (MP), devised for data communication between Control Stations and MicroBots, is described.

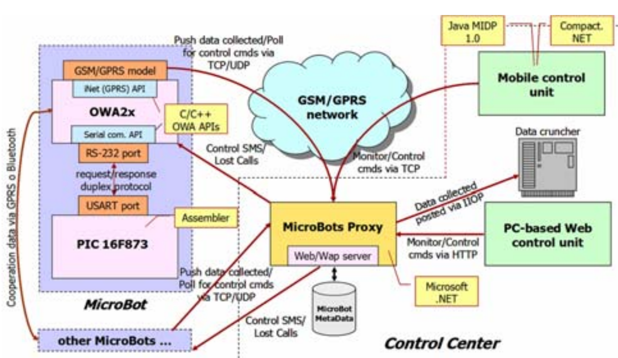

Fig. 2 The COMMBOTS Architecture.

A. MicroBots

Following the human body analogy, a MicroBot in COMMBOTS is composed of (see Fig.3):

- *Extremities*, i.e. the wheels together with the engines that give mobility to the MicroBots.
- *Senses*, i.e. sensors to capture information from its environment: luminosity, temperature, images, and so on.
- *Body*, i.e. the framework that gives weight and stability to the COMMBOT and acts as a container of its different hardware components.
- *Brain*, i.e. the microcontrollers that manage the engines, sensors and effectors bundled with the MicroBot, together with a communications module which allows data transmission with other robots and stations.

The logic system of the MicroBot is composed by two entities which coordinate its operation: a PIC16F873 [6] microcontroller, manufactured by MICROCHIP, and the owa22A communication module, manufactured by OWASYS [5]. The coordination between those two devices is achieved by a specially designed protocol, which exchanges 1 or 2 bytes long messages through the RS232 ports of both devices.

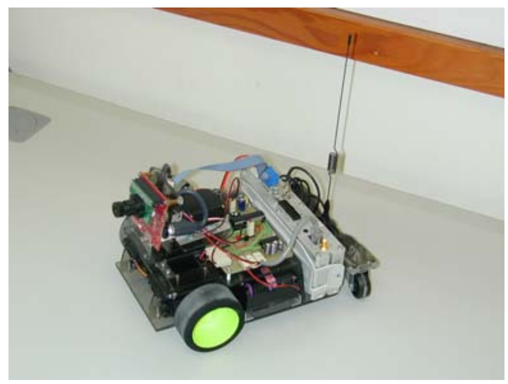

Fig. 3 A COMMBOT.

Given the limited processing power of the integrated microcontroller, the MicroBot's most CPU intensive tasks are undertaken by the owa22A. Such tasks are: (1) communication between the MicroBot and the MicroBots Proxy, (2) capture of digital images from the attached CMUcam2 camera [7] and their compression before delivery and (3) running the MicroBot behaviour programs. The microcontroller undertakes simple tasks that would unnecessary delay the owa22A module, such as control commands over the engines, sensors and effectors integrated in the MicroBot.

B. The MicroBots Proxy

The creation of this component which intermediates between MicroBots and Control Stations is justified by the following reasons:

- The MicroBots operate under battery. Thus, it is convenient to reduce their power consumption by avoiding having the communications module of the MicroBots actively listening to requests.
- Many telephony operators assign NAT (Network-Address-Translation) addresses to GPRS connected devices. Those IP addresses are not accessible outside the network operator's LAN.

- Frequently, the mobile operators raise firewalls and other obstacles to avoid connections to port 80 of GPRS mobile devices.
- Common functionality shared by both the static and mobile Control Stations can be factored out and placed within the business logic of a single component, namely the MicroBots Proxy. Thus, this component:
  - Provides a cache to avoid redundant communications with the remote MicroBots. This cache stores the latest sensor values of each MicroBot. Thus, the communication through GPRS will be reduced in one step, minimizing the cost and reducing the latency.
  - Ensures the fault tolerance of the MicroBot communications. Transparently to Control Stations, it is able to restart communications among itself and the MicroBots.

The MicroBots Proxy is a passive subject which never initiates connections to other components. The MicroBot communication module always initiates the connection with the proxy. Likewise, Control Stations also initiate connections. Between a MicroBot communication module and a proxy the following three connections are established:

1. *Connection for the reception of MP (MicroBot Protocol) commands and the delivery of responses*. Once the connection is opened, the software in the owa22A module blocks waiting for the arrival of commands.
2. *Connection for the delivery of captured images.* Once the MicroBot camera is activated, the owa22A module continuously transmits images to the proxy, where they are cached.
3. *Connection for the delivery of alerts.* Changes produced in the state of the MicroBot (temperature, luminosity, battery status, etc.) are notified to the MicroBots Proxy by means of this connection.

A mechanism employing control text messages and lost GSM calls has been created to prevent a MicroBot from having to keep a GPRS connection opened, with the associated waste of battery power, even when it is not being remotely controlled. The control text messages delivered activate/deactivate and parameter the connection opened with the MicroBot. Among other details, the longest period a connection can remain opened can be configured. Moreover, the owa22A modules connect/disconnect to the MicroBots Proxy when they receive a lost call from the proxy phone number. The SMS messages and lost calls are generated by a pair of web services available at [8].

Essentially, the MicroBots Proxy acts as a router of the requests arriving from the Control Stations towards the MicroBots and, at the same time, propagates the data generated by the MicroBots to the Control Stations. It has been implemented with Microsoft .NET, making extensive use of asynchronous sockets to efficiently control several MicroBots simultaneously.

C. Control Stations

The following varieties of Control Stations have been designed:

- The *Web/WAP Control Stations* transmit data through HTTP/WSP in XHTML or WML format. The interface of these stations is generated dynamically by the web component of the MicroBots Proxy. This web component uses the ASP.NET Mobile Web Controls framework [9], which adapts the mark-up pages generated to the target user agent (PC, PDA or mobile). Fig. 4 shows the same interface displayed on a web browser in a PDA, a mobile phone WAP browser, and a web browser in a PC, respectively. Given that the browser in a Pocket PC emits the HTTP `Accept-Encoding: gzip, deflate` header, i.e. it accepts XHTML content in compressed format, the MicroBots Proxy web component compresses the information before delivering it over GPRS.

Fig. 4 Interfaces WAP (mobile), web (PDA) and web (PC) for Control Stations.

- The *Pocket PC Control Stations* use TCP to transmit in binary MP commands to the MicroBots Proxy. This component has been implemented with Compact.NET [10], a .NET framework for the development of applications in PDAs. Fig. 5 shows, on its left hand side, the appearance of the interface for Pocket PC.
- The *J2ME Control Stations* run in every phone compatible with MIDP 1.0 [11], i.e., in the majority of the currently sold mobile phones. The communication with the proxy uses TCP sockets which transfer MP command and responses in binary format. Fig. 5 shows, on its right hand side, the J2ME interface.

Fig. 5 PocketPC and J2ME Interfaces for Control Stations.

D. The MP Protocol

The MicroBot Protocol (MP) governs the data flows exchanged between the MicroBots Proxy and the Control Stations, and between the owa22A modules and the proprietary Control Stations (those not using a web browser). Its design has followed a double objective: (1) use the smallest size messages and (2) avoid every confirmation or message communication not strictly necessary. This is to:

(1) incur in the minimum possible GPRS costs and (2) to reduce the required bandwidth as well as the communication latency.

MP follows the WBXML (WAP Binary XML) [12] principles to reduce the size of the messages. The command/response latency is improved by minimizing the number of messages delivered. A well known fact is that TCP in wireless environments behaves better with few bigger segments that many small ones [13]. TCP was designed for wired networks where latency, unlike the wireless case, is not a paramount factor. The MP commands are classified into three categories:

1. *Movement commands*.
   - Nine different movement actions (left forward, forward, right forward, left, stop, right, left backward, backward and right backward) are available.
   - The speed can be set to fast, normal or slow.
2. *Sensor commands*.
   - *Temperature*: commands to retrieve the current temperature and to set a notification alarm when the temperature moves outside a range. The user may activate the automatic temperature control by setting an upper and lower range for it.
   - *Luminosity*: commands to retrieve the current luminosity level and to set the automatic luminosity control.
   - *Anti-collision*: an MP command activates/deactivates the collision detection control, which prevents the MicroBot from colliding against walls and obstacles.
   - *Batteries*: commands to retrieve the current battery value and receive alerts when the value is under a threshold.
3. *Effector commands*.
   - *Digital camera*: commands to retrieve the images captured by the CMUcam2 [7] camera attached to the owa22A through RS-232, and to move it around. The MicroBot can move the camera up, left, centre, right and down.
   - *Lights*: commands to explicitly or implicitly (when the luminosity is below a threshold) switch on/off the lights of the MicroBot.

The following MP commands are sent between the Control Stations and the MicroBots Proxy, and between this latter one and the MicroBots:

- `MODULE, STATION`: it allows a client (module or station) to connect with the proxy. The telephone number of the module connecting or to connect to will be specified. For instance: `MODULE +34609421898`.
- `LIST`: it enumerates all the sensors and effectors available in a COMMBOT.
- `GET <control-element> [<param>(, <param>)*]`: indicates what sensor to obtain information from. For instance, `GET TEMP_LOW_VALUE` or `GET MV_MC_UP`.
- `SET <control-element> [<param>(, <param>)*]`: indicates the element to control and the values to assign. For instance, `SET TEMP_LOW_VALUE 20`.

On the other hand, the MicroBots respond to commands in the following format:

```
<response-code> <response-msg> CR-LF
Content-Length: <response-size> CR-LF
message-body
```

Noticeably, the MP format is very similar to HTTP [14]. However, the body of the message is delivered in binary format for performance reasons. It is not required to specify the MIME type of the response, because each request waits for a response in a predefined manner. Only the `Content-Length` header is required to facilitate processing. The `<response-code>` in the answer indicates whether the request was satisfied (`200 OK`) or not (`500 ERROR`).

MP is a very simple and easily extensible protocol. By means of the `LIST` message, it is easy to retrieve all the elements which can be configured with a `SET` command, and all the ones that can be queried with `GET`. The `LIST <control-element>` command enumerates all the parameter types allowed to configure and retrieve data from a control element.

IV. COMMBOTS PERFORMANCE RESULTS

Table 1 illustrates the average latencies encountered while issuing MP movement commands to a COMMBOT from three different Control Stations: (1) a MIDP 1.0 client running on a Nokia 6600, (2) the proprietary client running on a TSM 500 Pocket PC [15] and (3) the web browser of the TSM 500 PDA. Observe that the best results are obtained by the MIDP client, probably due to a more efficient implementation of the TCP/IP stack in the mobile phone than in the PDA. The worst results are obviously encountered for the web client. This is due to the fact that apart from the commands and their responses, it is necessary to deliver the mark-up of the pages to visualize. Moreover, the data transfer goes over HTTP, a level higher in the protocol stack than TCP (used by the proprietary clients).

TABLE I
DELIVERY OF MOVEMENT COMMANDS (50 BYTES) BETWEEN CONTROL STATIONS AND MICROBOTS

Device	Means (secs)	Standard Deviation (secs)
Mobile Phone MIDP 1.0 (TCP)	2.68	1.23
PDA Compact .NET (TCP)	4	1.67
PDA Web (HTTP)	5.54	1.36

On the other hand, Table 2 compares the latency encountered in the transfer of an image from the MicroBots Proxy to the proprietary Control Stations via TCP and to the web client through HTTP. Evidently, the best results correspond to the proprietary clients using the MP protocol over TCP rather than the less efficient HTTP.

TABLE II
DELIVERY OF MOVEMENT COMMANDS (50 BYTES) BETWEEN CONTROL STATIONS AND MICROBOTS

Device	Means (secs)	Standard Deviation (secs)
Mobile Phone MIDP 1.0 (TCP)	3.89	2.05
PDA Compact .NET (TCP)	3.724	2.722
PDA Web (HTTP)	6	1.91

V. THE CAREBOT CONCEPT

Disabled people experience many problems in their daily activities when interacting with the environment. Even basic actions become difficult for them. For example, determining who is ringing the doorbell, checking whether the washing machine has finished, or looking what has produced a noise at the corridor. Particularly some dangerous situations for common people can become even more risky for special collectives such as impaired or elderly people. For instance, a fire in the kitchen could be easily detected due to the generated smoke, if someone was watching; or the heating control is broken and it is getting colder in the room where an elderly person is sleeping; or maybe there is a noise at the entrance door and a disabled person cannot check whether something is going wrong.

Certainly, most of these problems may be solved by installing some kind of smart environment equipped with detectors, cameras, alarms, and so on. However, when those people move to another building or even a room not equipped with those devices, they are left by themselves.

The CareBot project proposes the design and implementation of a Sentient MicroBot to aid people with special needs. This device will check the ambient conditions surrounding the location of the people being aided and act in consequence in order to ease and enhance their daily activities. The CareBot will act autonomously, without demanding from the premises where it is located to be augmented with expensive and complicated home automation and sensorial facilities. Nevertheless, if that is the case it will be able of interacting with them.

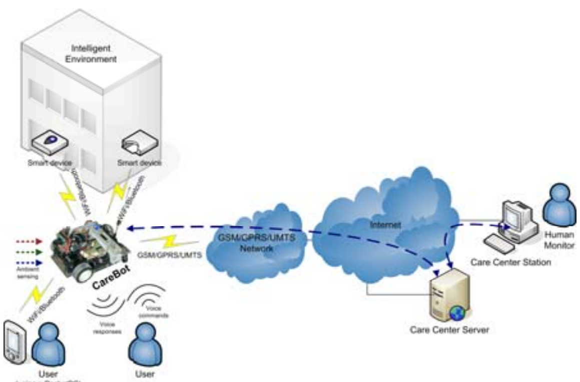

Fig. 6 The CareBot System.

The CareBot will be powered by many sensing devices. It will contain a colour camera that registers continuous images and visual activity in the environment, informing the patient or third parties interested (family members, doctors) about any detected anomaly. Apart from the camera, the microbot will also be equipped with other sensors that determine the temperature, collision, luminance, smoke, location or sound.

On the other hand, the CareBot will be equipped with a set of actuators which will enable it, among other things, to move around freely, turn on directional spot lights, use a mechanized arm to catch small objects, reproduce synthesized voice or communicate via cellular networks using a GSM/GPRS embedded module, or locally via WiFi or Bluetooth.

In essence, the CareBot will gather inputs through its sensors and act over its environment in consequence. The CareBot will follow the work pattern of a reactive system [16]. For example, it will monitor ambient conditions gathered by the different sensors in order to detect dangerous situations for the user, and so react to those situations accordingly. Depending on the detected risk and alert level, the CareBot may capture images, inform the user synthesizing voice, send the images via MMS, dial a phone number, send an SMS to a Care Center, or even interact with another system to create a coordinated response.

Although not strictly necessary, the CareBot will benefit from AmI environments populated by smart interactive objects. The combination of the CareBot capabilities and AmI environments will allow the former to develop complex behaviours and the latter environment adaptations always for the user's sake. For instance, the CareBot will not only determine that the temperature is too cold, but it will also negotiate with the heating control system a new temperature more suitable for the people being taken care of. In the case that negotiation cannot be performed successfully, or the desired temperature cannot be reached, the CareBot may decide to launch an alarm action such as advising the user via the synthesized voice or informing the associated Care Center or the relatives of the user.

The CareBot may act autonomously following a set of predefined or learned behavioural rules, or its operation may be supervised through a PDA with a friendly user-interface by the disabled or elderly person. The user may even use natural mechanisms such as voice to issue commands over the microbot, and receive responses in the form of synthesized voice. Furthermore, given that the CareBot will be equipped with a GSM/GPRS module, it is even possible to monitor the activities of the CareBot and oversee the person with special needs remotely from a Care Center by means of a web interface (equipped with the appropriate authorization and security measures).

For the CareBot to be able of acting autonomously it will need to move freely around the premises where the disabled or elderly person is located. One of our goals is not to tie the CareBot to a given location (house, building). Thus, the CareBot will need to find its whereabouts autonomously. For that we propose to use a visual marker-based recognition system with very low processing demands, such as TRIP [17]. Any environment could easily tagged by means of barcodes and arrows that indicate directions towards the front door, the exit of a room and the default location of the disabled or elderly person, or simply the current location of the CareBot. We believe that a system of this nature imposes hardly any extra infrastructural costs, since it will use the CareBot camera and communication module processing power to recognize those easily printable barcodes or makers. In addition, the CareBot could download maps of the current location of the disabled or elderly person that would complement and give sense to the markers scattered through the skirting boards of those premises. All these mechanisms turn the CareBot into a full assistant for disabled or elderly people as well as a valuable partner. Analogously to the CareBot case, we have concluded some work on which mobile phones played the role of intermediaries between us and the environment [18].

Fig. 6 depicts diagrammatically how the CareBot may interact with the local environment, and how the person with special needs or a remote supervisor may monitor and control de CareBot.

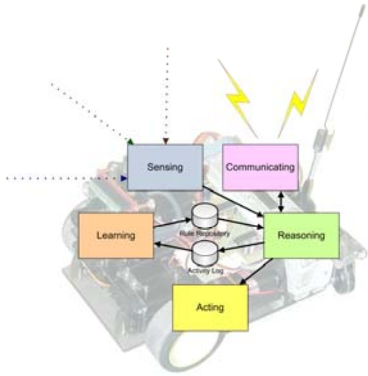

Fig. 7 The CareBot Internal Software Modules.

VI. RELATED WORK

The PocketCERO [19] project has devised PDA interfaces that allow service robots, those that assist people with special needs (e.g. elderly people), control the elements in their environment. The COMMBOTS system was thought with a more industrial purpose in mind. However, our current work on the CareBot concept approaches more to the PocketCERO concept.

The special requirements of mobile communications with robots have been studied thoroughly by other authors [20], which have also considered the use of WAP-enabled phones for robot telemetry [21]. We have followed their guidelines in the design of our MP protocol.

The KDDI mobile operator has also considered the combination of mobile phones and microbots with Pirkus [22], a Bluetooth mobile phone controlled robot. In COMMBOTS, we use GPRS instead, since it gives us global access to the microbots, independently of our location. A similar approach was previously followed by Fujitsu's Maron [23], an internet-enabled home robot capable of undertaking remote household monitoring, remote control of household appliances, or to serve as a hands-free phone.

VII. CONCLUSIONS AND FURTHER WORK

The implementation of the COMMBOTS architecture has allowed us to study the feasibility of achieving real-time control of remote devices by means of GPRS. The results obtained have shown that the remote control through web/WAP from a mobile station is not feasible when real-time responses are required. Hopefully, when UMTS data connections become the norm this issue will be solved. However, the design of a protocol specially catered for wireless communication together with the optimizations and caches of the central component in our 3-tier architecture, namely the MicroBots Proxy, have shown that the real-time control of remote devices by means of GPRS is possible from wireless Control Stations. The results and source code of the COMMBOTS system are publicly available at [24].

Currently, we are working on an implementation of the CareBot concept following the design shown in Fig. 7. The CareBot will not only be equipped with sensing, communicating and interaction modules as COMMBOTS, but it will also incorporate a reasoning module based on ECA rules and a learning module, to significantly increase its degree of autonomy. It will also offer more natural interaction mechanisms (voice, gestures) as we have done with our MobileSense platform [18].

REFERENCES

[1] N. Shadbolt, "Ambient Intelligence". IEEE Intelligent Systems, July/August 2003 (2003) 2-3

[2] M. Weiser. "The Computer for the 21st Century". Scientific American, (1992) vol. 265, no. 3, pp. 94-104.

[3] "When Robots Rule the World", http://www.wired.com/news/technology/0,1282,65408,00.html, Wired News, October 2004

[4] "Mars Exploration Rover Mission", http://marsrovers.jpl.nasa.gov/home, NASA, 2004

[5] OWASYS owa2X M2M modules, http://www.owasys.com/en/BOK_100_1000_R3D.pdf, 2004

[6] "Microchip PIC 16F87X Data Sheet", http://ww1.microchip.com/ downloads/en/DeviceDoc/ 30292c.pdf, Microchip Technology, 2001

[7] "The CMUcam2", http://www-2.cs.cmu.edu/~cmucam2/, Robotics Institute, Carnegie Mellon University, 2004

[8] "Cátedra de Telefónica Móviles de la Universidad de Deusto", http://www.ctme.deusto.es, 2005

[9] "Microsoft ASP.NET Home Page", http://asp.net, Microsoft Corporation, 2005-01-31

[10] "Mobile Application Development Toolkit", http://www.microsoft.com/downloads/details.aspx?FamilyID=f432833 3-0fd4-4348-88c0-39d10fb64f0a&DisplayLang=en, Microsoft Corporation, 2004

[11] "Java 2 Platform, Micro Edition (J2ME)", http://java.sun.com/j2me/, Sun Microsystems, 2005

[12] "WAP Binary XML Content Format", http://www.w3.org/TR/wbxml/, W3C, June 1999

[13] G. Xylomenos, G. Polyzos and P. Mahonen, "TCP Performance Issues over Wireless Links", IEEE Communications Magazine, April 2001

[14] "Hypertext Transfer Protocol", http://www.w3.org/Protocols/, W3C, 2004

[15] "TSM 500", Telefónica Móviles, 2005 http://www.movistar.com/ particulares/renueva/data/tsm/tsm500.htm

[16] D. López de Ipiña et al. "An ECA Rule-Matching Service for Simpler Development of Reactive Applications", Published as a supplement to the Proceedings of Middleware 2001 at IEEE Distributed Systems Online, Vol. 2, No. 7, November 2001

[17] D. López de Ipiña, P. Mendonça. and A. Hopper, "TRIP: a Low-Cost Vision-Based Location System for Ubiquitous Computing", Personal and Ubiquitous Computing journal, Springer, vol. 6, no. 3, May 2002, pp. 206-219

[18] D. López de Ipiña, I. Vázquez, J. Ruiz de Garibay and D. Sainz, "Interacting with our Environment through Sentient Mobile Phones", to be published in Proceedings of 2nd International Workshop on Ubiquitous Computing (IWUC-2005), May 2005 .

[19] Hüttenrauch H. and Norman M, "PocketCERO – Mobile Interfaces for Service Robots", Proceedings of Mobile HCI 2001: Third International Workshop on Human Computer Interaction with Mobile Devices, Dunlop, September 2001

[20] Gage, D.W, "Network Protocols for Mobile Robot Systems", SPIE Proceedings, 3210. Presented at Mobile Robots XII, Pittsburgh, PA, 14-17 October, 1997

[21] Sgouros N. and Gerogiannakis S., "Integrating WAP-Based Wireless Devices in Robot Teleoperation Environments", Proceedings of IEEE ICRA 2002, pp. 1191-1196, ISBN 0-7803-7273-5, 2002

[22] KDDI, "Pirkus: Bluetooth Mobile Phone Controlled Robot", http://www.blueserker.com/html/modules.php?op=modload&name=News&file=article&sid=290&mode=thread&order=0&thold=0, 2005

[23] Fujitsu, "PFU Launch Initial Sales of MARON-1 Internet-Enabled Home Robot to Solutions Providers in Japan Market", http://pr.fujitsu.com/en/news/2003/03/13.html, 2003

[24] "COMMBOTS Homepage", http://www.ctme.deusto.es/ COMMBOTS, Universidad de Deusto, 2005