HARDWARE INDEPENDENT ARCHITECTURE FOR AUTONOMOUS COLABORATIVE AGENTS

Guillermo Glez. de Rivera, Ricardo Ribalda, Kostadin Koroutchev, José Colás, Javier Garrido
Escuela Politécnica Superior. Universidad Autónoma de Madrid. C/ Francisco Tomás y Valiente 11, Madrid, Spain
Email guillermo.gdrivera@uam.es, ricardo.ribalda@uam.es, jose.colas@uam.es, javier.garrido@uam.es

Keywords: Mobile robot, XML/RPC, WebService, Wireless.

Abstract: We describe a modular mobile robot test system. This architecture allows easy inclusion of user hardware and communication modules. A client-server, XML/RPC based approach makes the system easy to program and neutral in respect to the operating system and the programming language used. The hardware modules are included using a hardware independent protocol. This feature of the system makes it very flexible and easy to use and reconfigure. The architecture by itself has support for many different communication modalities.

1 INTRODUCTION

Nowadays there is a great demand of robotic systems designed to solve complex tasks in fields as manufacture, construction, transport, medicine and others (Sybley 2002). Furthermore, in web-controlled systems, robots play the role of a physical mediator, enabling remote operators to acquire information, explore, manipulate, communicate, and establish long-range interactions with other persons (Wang, 2004), (Wang, 2003).

During the development process of a robotic project, two different teams are involved, namely hardware and software group. The two teams interact each other in order to solve the problem. In many cases, the communication between the teams is not possible and therefore, each group limits its activities to the basic function only. For example, a frequent practice for the software team is to limit its efforts to simulated environment.

This paper presents a flexible and generic platform that serves to the hardware designers as a full operative open system, where they can include easily their own devices (see also Glez. De Rivera 2002). The only requirement imposed on the devices is to comply with relatively simple communication protocol.

The final result is, on one side, a mobile robot equipped with sensors, actuators and different communication interfaces, and on the other side, software that controls the robot's sensors-actuators. The communication interfaces allow robot-robot and robot-control centre collaboration.

The software design includes as much drivers as necessary to handle the sensors and the actuators. Also, the necessary communication protocols are defined, having in mind a variety of physical communication media.

Multiple network interfaces are included, which allow the agent to choose the optimal media in each particular application. For the end user of the platform, the system looks like a "black box" with which he interacts through simple XML-RPC calls.

XML-RPC (XML-RPC, 2000), (Dissanaike 2004) allows a software running in different environments to make operating system independent procedure calls over TCP/IP. This makes possible to use different hardware elements, as for example sensors and actuators to be used from most platform/programming languages. From the programmer point of view, the whole robot is converted into a simple set of RPC functions.

This paper describes the global architecture of the platform and its basic hardware structure. Similar systems are described in (Golovinski, 2004), (Hoopes, 2003) and (Navarro 2002). The system described in this paper, using client-server paradigm, XML/RPC and machine/language independent way to introduce hardware/software modules in assumed to be easier to program and reconfigure.

Using this architecture, a variety of peripheries, as for example video camera, microphones, speakers have been tested and are in test process.

2 ROBOT ARCHITECTURE

A scheme of the system is showed in Figure 1. Basically, each robot acts as a server. A set of clients can query and command the robots. The clients can be installed in the robots itself. The communication server-server or client-server, can be done using Internet or another type of private network.

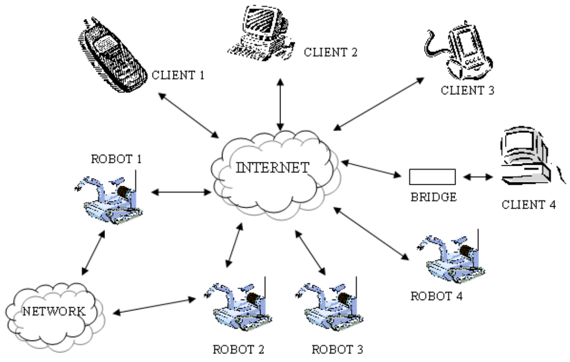

Figure 1: System architecture

If given client does not support some of the conventional data physical transport media, the platform allows to use a bridge.

The client can be any device that supports TCP/IP, like PC, PDA, mobile telephone, embedded system, etc.

Considering the robot as hardware architecture, the robot is a mechanical structure with traction system, set of sensors e.i. ultrasounds/infrared proximity sensors, end-race switches, camcorder, microphone, etc.. All components are controlled by a microprocessor that also supports the external communications.

It is important to point out the flexibility of the hardware structure, derived from:

a) The simple way to add new devices. The designed software platform allows changing or adding any hardware element with USB connection.

b) The possibility of establishing a communication with the robots by means of different physical media. Due to the capacity of the robots to choose the most suitable media for each situation and the ability to use redundancy of the media, the reliability of the communications is high.

c) The control and data acquisition is possible from any devices that have Internet connection, suitable client application and adequate permissions.

The system has a set of predefined control primitives to facilitate to the user low level operation as for example, to advance straight, left or right, to take and send a video frame, and similar.

The aim of the platform is to offer to the final user a hardware solution for implementing any application or algorithm. The platform offers to the user a real hardware system that he can manage as call functions to an especial software library.

These functions are defined in any part of the code and the arguments of then will be for example the robot IP address, identifying the physical communication media chosen, or any parameter to select a determinate sensor system predefined in the robot.

2.1 Physical structure

The chassis is formed with aluminium profiles forming rigid enough structure that allow fixing of all necessary elements.

The traction mechanism was chosen to be a system of rubber wheels, two driving rubber wheels and a third rubber "crazy" wheel.

Each driving wheel is connected to a powerful step motor that allows fine control of the robot position.

In order to obtain that fine control and to release the main controller from this time-consuming task, the motors are handled by dedicated microprocessor (PMD 2002) and run their own specially designed Linux driver. The access to this driver is made by means of USB protocol.

2.2 Central control system

The central control system is based on the well-known PC architecture. The motherboard is a VIA EPIA M10000 (VIA 2000), based on the microprocessor Via C3/EDEN and running under Linux operating system.

The communications are implemented using large number of network elements, such as Ethernet, wireless, parallel and serial ports, USB, bluetooth, GSM/GPRS and radio in the band of 433MHz or 868MHz. Wireless is designed as principal connection mode, because is the one that better adjusts most of the necessities. The other methods have been included in order to improve and generalize the system as well as to add communication redundancy.

The internal connections of different elements, such as sensors or actuators, use exclusively USB ports, because this interface is supported by the most of peripheral devices. USB devices are low cost and they do not need any additional power source.

A block diagram of the control structure can be observed in Figure 2.

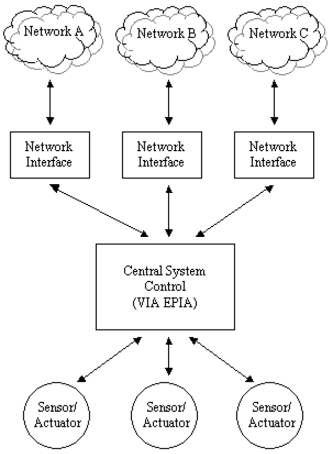

Figure 2: Block diagram of control structure

2.3 Robot state manager

This hardware element is mounted on the structure of the robot and allows independent real-time monitoring of the robot's state, as for example the current communication physical media, the state and values of the connected sensors, the battery state, power consumption of different modules, etc.

The state manager is implemented using the GP_Bot Platform (Glez. de Rivera, 2002) connected to the mainboard by serial port. Data are displayed in a 128x64 graphic LCD screen (Ampire AG-12864EYIQY-00). Screen menus are accessed by means of a button set.

2.4 Sensorial server

A platform based just on PC is not flexible enough for all propose as Input/Output processing. So we developed a method to upgrade it introducing external elements in a simple and standard way. The external element, normally, will be composed by a control element with sensors/actuators. In some cases a specific sensor needs more attention of the CPU. In these cases small autonomous control systems, named sensorial servers (SS), are designed. The SS are based on a microcontroller that attends the sensor-effectors demands without occupying the central unit (Figure 3).

In order to standardize the operation of the sensorial servers, a communication protocol with the central control has been defined.

In each one of these servers, an application that attend demands of the central control is running. The connection between the control central and the server is made trough an USB port.

Most of the sensors/actuators are designed for working using serial communications like RS232, but in the other hand, a normal PC have just up to 4 serial ports. To solve this we propose that the RS232 communication goes over a USB communication using USB UARTS. They support speeds up to 3M bauds, what is enough for most applications. Also, because they work through USB interface, up to 127 sensorial servers can be connected for each USB port.

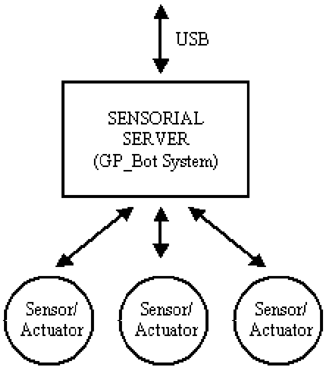

Figure 3: Block diagram of the sensorial server

In general, there are two ways to connect SS: developing a protocol for each particular SS or developing a standard SS protocol. If we choose the former way, the hardware expert will have to develop a driver and its interface. The later method gives clear advantages in the development process and makes the programs more simple and uniform. We restrict the parameters of interface calls to several primitive types, as integers and strings in order to make the interface as simple as possible. One of the reasons to do so is because at least some of the functions of the interface have to be implemented in hardware.

The communications with SS appear to the central unit as a set of function calls. In each functions a set of input and output arguments are defined.

One important condition of these protocols is the reflection activity; that means that the protocol have to list all the defined functions supported by the SS. With this capacity, when a new SS is connected to the main control, this control or any client connected to the robot trough the net, can know and use them.

In the basic robot architecture a set of predefined SS, based on the previous mentioned GP_Bot platform is included.

Due to the open design of the platform, any user can include any kind of new sensor, directly o trough a microprocessor based server with the only condition of to be adapted to the previous declared protocol.

2.5 Speech synthesis and recognition

The robot's architecture includes a set of microphone and speaker, as speech system sensor for two main reasons:

As a remote speech server, to record and transmit sounds produced in the robot's surroundings and to reproduce speech messages received trough the net.

As an element for advanced research in the area of human computer interfaces, to recognize human speech orders, and to synthesize robot speech answers.

3 CONCLUSIONS

As final result of this work a set of robots has been designed. One of them is showed in the Fig. 4a & 4b, has the following characteristics:

a) Main control: mainboard VIA EPIA M10000
b) Sensorial servers: GP_Bot platform
c) Network interfaces: Wireless, Ethernet, USB
d) Two stepping motors (SST58D3820 model), controlled by a motor processor (PDM).
e) Four Infrared sensor (Sharp GPD2D12).
f) Two Ultrasound sensor (SRF04).

We have developed new drivers for other types of sensors as pyrometers, wet sensors or gas sensors.

Figure 4a. Image of the robot

Figure 4b: Image of the robot without top

4 REFERENCES

Dissanaike S.; P. Wijkman and M. Wijkman, 2004. *Utilizing XML-RPC or SOAP on an embedded system,* Proc. 24th International Conference on Distributed Computing Systems Workshops, pp438–440.

Glez. de Rivera G. et al.. 2002. *GP_BOT: Plataforma Hardware para la enseñanza de Robótica en Ing. Informática*. In (TAEE'02). pp67-70. UPGC.

Golovinski A., Yim M., Zhang Y., Eldershaw C., Duff D., 2004. *PolyBot and PolyKinetic System: A modulR Robotic Platform for education*, International Conference on Robotic & Automation.

Hoopes D., Davis T., Norman K., Helps R., 2003, *An Autonomous Mobile Robot development platform for teaching a graduate level mechatronics course*. 33rd ASEE/IEEE Frontiers in Education Conference.

Navarro L., et al., 2002. *Millibots. The development of a framework and algorithms for a distributed heterogeneous robot team.* In IEEE Robotic and Automation Magazine, vol 2, no 4, pp 31-40.

PDM-MC3410 Pilot Motion Processor

Sibley G.T., Rahimi M.H., Sukhatme G.S., 2002. *Robomote : a tiny mobile robot platform for large-scale ad-hocsensor networks*. IEEE, Robotics and Automation. ICRA'02, pp 1143-1148.

VIA Technologies Inc, http://www.viavpsd.com/product/

Wang D.; X. Ma and X. Dai, 2004. *Web-based robotic control system with flexible framework.* Proc. ICRA '04. IEEE International Conference on Robotics and Automation, Volume: 4, pp:3351-3356.

Wang X.; M. Moallem and R.V. Patel. 2003. *An Internet-based distributed multiple-telerobot system.* IEEE Transactions on Systems, Man and Cybernetics, Part A, Vol:33, Issue: 5, Sept. 2003, pp:627 – 634.

XML-RPC http://www.xmlrpc.com/