International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 3 – No.6, July 2012 – www.ijais.org

Multi-Agents Systems for Remote Control on Internet

Adil SAYOUTI
Team Architecture of Systems
LISER - Laboratory
ENSEM, Hassan II University
BP 8118, Oasis, Casablanca, Morocco

Hicham MEDROMI
Team Architecture of Systems
LISER - Laboratory
ENSEM, Hassan II University
BP 8118, Oasis, Casablanca, Morocco

ABSTRACT
Computer science spans a range of topics from theoretical studies of algorithms and the limits of computation to the practical issues of implementing computing systems in hardware and software, this discipline contain the treatment of theory of computation, algorithms and data structures, parallel computation, computer networking and remote control over Internet of distributed intelligent systems which is our area of research. This technology evolves with a high rapidity with an aim to make the systems controlled autonomous and intelligent. To fulfil these requirements, we decided to use a multi-agents formalism that fits naturally our needs. In a first part of this paper, we are presenting the main interests of such a remote control and we are describing the characteristics of agents in order to validate our choice based on the agent approach. In a third part, we are proposing a system architecture to realize such remote control over Internet. In the fourth part, we are presenting our control architecture. In fifth part, we are describing the proposed software architecture. An illustration of our architecture is given in an application of control of an autonomous mobile robot.

Keywords
System Architecture, Control Architecture, Multi-Agents System, Distributed System, Internet, QOS, Web Application.

1. INTRODUCTION
The Internet has become part of everyone's daily life. It connects people worldwide and provides a means of affordable communication. In order to take advantage of this ever popular medium, researchers have started to use it as a medium for remote control of a robot. Such remote operation can replace the need of hurnan presence at hazardous or unreachable location The use of Internet gets important advantages. It's cheaper than high-quality support and available all-round the world. Today from all points of the earth, it is possible to be connected to Internet.

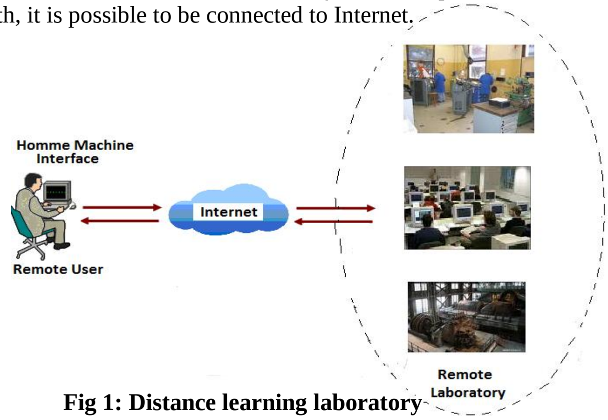

Fig 1: Distance learning laboratory

This Worldwide implementation permits to use this technology in new fields of activity: Telesurgery, Telemedicine, Teleteaching (fig. 1), Telemanufacturing [1], Telemaintenance…

But the use of Internet presents an inconvenient. There is no Quality of Service (QoS) on this network support. As security cannot be provided by the network, it has to be controlled by the architecture of the system. The organization of a robotic system - or its control architecture - determines its capacities to achieve tasks and to react to events. The control architecture of an autonomous robot must have both decision making and reactive capabilities: situations must be anticipated and the adequate actions decided by the robot accordingly, tasks must be instantiated and refined at execution time according to the actual context, and the robot must react in a timely fashion to events. This can be defined as a rational behavior, measured by the robot's effectiveness and robustness in carrying out tasks. To meet this global requirement, a robot control architecture should have the following properties: Intelligence, Autonomy, adaptability, Reactivity, Robustness and Extensibility. To fulfil these requirements, we decided to use a multi-agents formalism that fits naturally our needs. Since their coming out in the 80's multi-agents systems have been considered as "societies of agents", i.e. as a set of intelligent and autonomous agents that interact together to coordinate their behaviour and often cooperate to achieve some collective goal.

A characteristic is an intrinsic or physical property of an agent. The following are the agent characteristics [2]:

- Autonomy: An agent has a degree of autonomy. An agent has some states (not available to other agents and components system).
- Situated: An agent is situated in its environment (physical or virtual).
  An agent has a representation of its environment.
- Reactive: An agent may perceive their environment through sensors. An agent can act on its environment through effectors.
- Social: An agent is able to interact and communicate with other agents (by language of communication). An agent is able to cooperate to solve problems or make tasks.
- Proactive: An agent is able to "take the initiative" to achieve its purpose or perform tasks (and adopt the appropriate behavior).
- Active: An agent is always active. It therefore necessarily run in a thread or independent process.

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 3 – No.6, July 2012 – www.ijais.org

- Learning: An agent is able to learn and evolve in response to learning. An agent is capable of changing behavior (depending on experience past).

This article is presented as follows: on the next section, we describe some projects of remote control of robotic systems. Then we will propose a system architecture to realize such control over Internet. In section 4, we describe our control architecture based on multi-agents systems to take into account the lack of quality of services of Internet. In section 5, We present the proposed remote control software architecture. Then we present an application of control of an autonomous mobile robot (Khepera Mobile Robot) as an illustrative example. Finally, some conclusions are presented in section 7.

2. REMOTE CONTROL – STATE OF THE ART

During the Nineties, several projects appeared of robotic systems control, using Internet as communication network [3] with various objectives.

The Mercury Project [4] is believed to be first system that allowed Internet users to remotely control robotics via Internet. This project launched the first system allows users to alter the real world. This project is initialized by an interdisciplinary team of anthropologists and computer scientists. They want to explore an area dubbed “Mercury site”. Because nobody can work in this dangerous area, the remote robotics is a good choice. The remote control of the robot is designed to excavate the surface with short burst of compressed air and then the surface is revealed and the relevant data can be collected by the anthropologists. After the success of the initial exploration, the site is open to all the researchers who are interested in having a remote control of the robot via Internet. The successes of Mercury Project is not only on its excavation purpose, but also showed the possibility of control the robot via Internet. This is the milestone on Internet telerobotics, more and more Internet telerobotics projects were launched in the later years.

Telegarden is the second Internet telerobotics from Goldberg and al. This Telegarden system allows Internet users to view and interact with a remote garden filled with living plants. Users can plant, water, and monitor the progress of seedlings via the tender movements of an industrial robot arm.

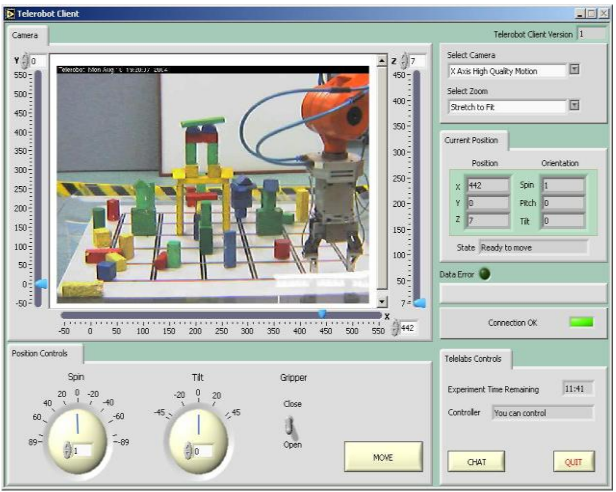

Fig 2: Australian Telerobot on the Web

The Australian Telerobot on the Web project [5] is established by Taylor at The University of Western Australia. A six degree of freedom robotic arm is controlled through Taylor’s Internet telerobotics system (fig. 2). The robotic arm can play the wooden block on a table. The user connects the server via Internet connection, and can log to the system with an identification check. This project is very successfully due to it’s an interesting wooden block game as well as users can play it via Internet.

The launch of RHINO project [6] indicates the possible potential of Internet robotics in daily life. RHINO project is initially launched for a museum of contemporary technology in Bonn Germany. Visitors of the "Deutsches Museum Bonn" will have the opportunity to be shown through the museum by a mobile robot “RHINO”. RHINO can provide the user with the information they concerned as well as more information in deep upon request.

The research focus of Xavier [7] is to study the local intelligence of the robot as well as users’ interface. The research team has considered the supervisory control which aims to give robot the command at a higher level. This scheme can reduce the influence of Internet time delay, but prevents more interaction between users and robot. In this situation, users can’t interact with the robot immediately. Remote users need fast feedback (image) when controlling the robots on the web facing unpredictable Internet time delay (limited bandwidth). The supervisory control can indeed reduce the bandwidth requirement, but at the same time reduces interactivity. These existed problems are highly concerned in the later years’ research of Internet telerobotics.

The PumaPaint project [8] came out from the collaboration between University of Wisconsin and Wilkes University. The computer science department at UW needed a PUMA robot as undergraduate teaching resource. But if they develop and maintain the system themselves, it’s money and time costly. So, they decide to share the installed one at Wilkes University. Students from University of Wisconsin can access the PUMA robot via Internet connection. The system is developed in Java considering the cross platform advantage of java as well as the reusability of Java program. The Java Virtual Machines (JVM) is involved in the development, every machine need to install this before using the system, but the JVM is quite popular in most of web browsers.

The possibility of piloting a Khepera robot [9] was made available to the general public over the web in December 1996 (KhepOnTheWeb). By means of a WEB client (Netscape), the user can move forward or turn the robot and receive images of the remote environment (fig. 3). He can also choose the point of view and receive images either from the robot's on-board camera or from a camera mounted on the ceiling.

Fig 3: The control interface – khep on the web

All these experiences are really interesting because they have treated the problem of remote control in different manners and in different contexts. To develop safe and evaluated web-based remote control, one has to take into account all of them. Nevertheless, we can notice that none of these works have tried to develop some control architecture and the unpredictable nature of the Internet is not really taking into account with all the consequences. In the next sections, we will propose some solutions to these two problems.

3. PROPOSED SYSTEM ARCHITECTURE

With the rapid growth of the Internet there are many communications technologies available to execute requests in a networked environment. Currently the most widely used web browser is the Hypertext Transfer Protocol (HTTP). It can be executed with the Communication Gateway Interface (CGI) for remote control, which is one of methods used in many web-based telerobot systems [10]. Through the Hyper Text Markup Language (HTML) form, a request can be passed from client to server to launch a process to perform some predetermined actions in the server. A dynamically generated HTML page will return the results to the client. But CGI has a number of limitations [11] such as its slow response speed. Moreover, a complete HTML page must be generated with each request while the resulting page is still static. So it is not suitable for real-time remote control. In contrast, Java provides the capability to implement network connections and thus avoids the limitations of CGI. A Java applet can operate within the browser and hence is accessible by most computers on the Internet [12]. Rather than being static, a Java applet also enables an interface to dynamically change its content due to the fact that the Java applet is an executable within a web page.

From the study of the experiences made on Internet, a common frame (see fig. 4) can be described about the operational aspects of a remote control application [13]. The remote user, through his Internet navigator, addresses a request to a Web server (step 1) and downloads an application on his work station such as for example an applet Java (step 2). A connection is then established towards the server in charge of the management of the robot to control (step 3). The user is then able to take the remote control of it. In parallel to step 3, other connections are also established towards multi-media servers broadcasting signals (video, sound) of the robot to be controlled.

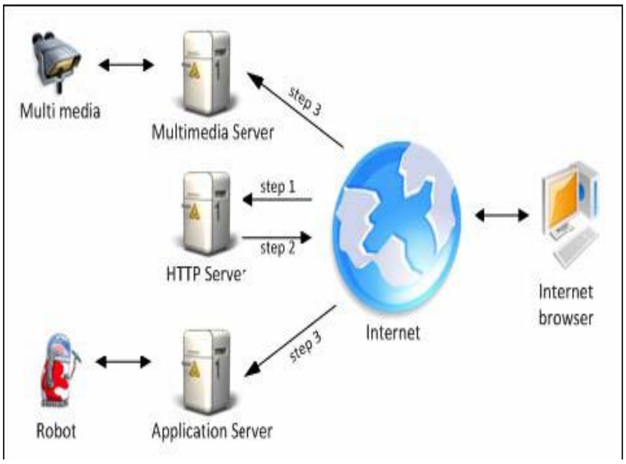

Fig 4: System architecture proposed in remote control

4. THE CONTROL ARCHITECTURE

When turning a robot on, the problem of its autonomy is quickly addressed. However, several types of autonomies can be considered: energetic autonomy, the behaviour autonomy or smart autonomy. The designer has to choose the way he will give autonomy to his robot. He has mainly two orientations: "reactive" capacities or "deliberative" capacities [14]. These two capacities are complementary to let a robot perform a task autonomously. The designer must built a coherent assembly of various functions achieving these capacities. This is the role of the control architecture of the robot. To design an autonomous robot implies to design a control architecture, with its elements, its definitions and/or its rules.

4.1 Brief Overviews of Control Architecture

One of the first author who expressed the need for a control architecture was R.A. Brooks [15]. In 1986, he presented an architecture for autonomous robots called "subsumption architecture". It was made up of various levels which fulfil separately precise function, processing data from sensors in order to control the actuators with a notion of priority. It is a reactive architecture in the sense that there is a direct link between the sensors and the actuators. This architecture has the advantage to be simple and thus easy to implement, nevertheless, the priorities given between the different actions to perform are fixed in time and do not allow an important flexibility.

Then other various architectures were developed based on different approaches, generally conditioned by the specific robot application that the architecture had to control.

The architecture 4-D/RCS developed by the Army Research Laboratory [16] has the main characteristic to be made up of multiple calculative nodes called RCS (Real time Control System). Each node contains four elements, performing the four following functionalities: Sensory Processing, World Modeling, Behavior Generation and Value Judgment. Some nodes contribute to the perception, others contribute to the planning and control. These nodes are structured in levels, in which one can find the influence of the reactive behaviors in the lower levels and of the deliberative behaviors in the higher levels.

The Jet Propulsion Laboratory developed in collaboration with NASA its own control architecture called CLARAty [17]. Its principal characteristic is to free itself from the traditional diagram on three levels (Functional, Executive, Path-Planner) and to develop a solution with only two levels which represent the functional level and the decisional level. A specific axis integrates the concept of granularity of the architecture for compensating the difficulties of understanding due to the reduction of the number of levels. One of the interests of this representation is to work at the decisional level only on one model emanating from the functional level. The decomposition in objects of this functional level is described by UML formalism (Unified Modeling Language) that allows an easier realization of the decisional level.

The LAAS architecture (Laas Architecture for Autonomous System) [18] is made up of three levels: decisional, executive and functional. Its goal is to homogenize the whole mobile robotics developments and to be able to re-use already designed modules. All the modules of the functional level are encapsulated in a module automatically generated by GenoM. These have to interact directly with the actuators and other modules of the functional level. The higher level is a controller of execution (Request & Ressources Checker). Its main function is to manage the various requests emitted by the functional level or the decisional level. The operator acts only at the decisional level by emitting missions which depend on the information incoming from the lower levels. This architecture has an important modularity even if the final behavior is related to the programming of the controller of execution.

A. Dalgalarrondo [19] from the DGA/CTA proposed another architecture. It presents a hybrid control architecture including four modules: perception, action, attention manager and behavior selector. It is based on sensor based behaviors chosen by a behavior selector. The “perception” module carries out models using processing which are activated or inhibited by the “attention manager” module. The “action” module consists of a set of behaviors controlling the robot effectors. A loop is carried out with the information collected by the perception part. This is particularly necessary for low level actions. The “attention manager” module is the organizer of the control architecture: it checks the validity of the models, the occurrence of new facts in the environment, the various processing in progress and finally the use of the processing resources. The “behaviour selector” module must choose the robot behavior according to all information available and necessary to this choice: the fixed goal, the action in progress, representations available as well as the temporal validity of information.

The DAMN architecture (Distributed Architecture for Mobile Navigation) results from work undertaken at the Carnegie Mellon University. Its development was a response to navigation problems. The principle is as follows: multiple modules share simultaneously the robot control by sending votes which are combined according to a system of weight attribution. Then, the architecture makes a choice of controls to send the robot, by a fusion of the possible solutions [20].

This architecture proposes the dominating presence of only one module which decides the procedure to follow. This forces to concentrate all the evolution capabilities of the robot. This mode of control does not make it possible to understand or anticipate the probable behavior of the robot with respect to an unexpected situation.

4.2 The Proposed control architecture

Our architecture is based on the same architectures principles that have been presented in the previous section. It relies on the concept of levels initially developed by R. Brooks and which appear in architectures proposed by AuRA or LAAS. We propose a hybrid control architecture which combines aspects of classic control and behavior-based control. Our architecture called EAAS for EAS Architecture for Autonomous system [21], including a deliberative part (Actions Selection Agent) and a reactive part. It is made up of two parts, each using distinct method to solve problems (see fig. 5). The deliberative part which uses methods of artificial intelligence contains a path planner, a navigator and a pilot. The reactive part is based on direct link between the sensors (Perception Agent) and the effectors (Action Agent). Fundamental capacities of our architecture encompass autonomy, intelligence, modularity, encapsulation, scability and parallel execution. To fulfil these requirements, we decided to use a multi-agents formalism [22] that fits naturally our needs.

The multi-agents system paradigm is one of the most promising approaches to create autonomous, open and dynamic systems, where heterogeneous entities are naturally represented as interacting autonomous agents, which can enter or leave the system at will. Interaction among autonomous agents is fundamental to the dynamic of multi-agents systems. Agents need to interact and coordinate their activity to carry out their common global goal. The development of multi-agents systems is mainly due to its interactions with different scientific domains, in particular with biology. Biology, and especially ethology, inspired the first architecture and several distributed algorithms [23], [24].

The communication between agents in our architecture is realized by messages. Object oriented language is therefore absolutely suited for programming agents (we chose java). We use threads to obtain parallelism (each agent is represented by a thread in the overall process).

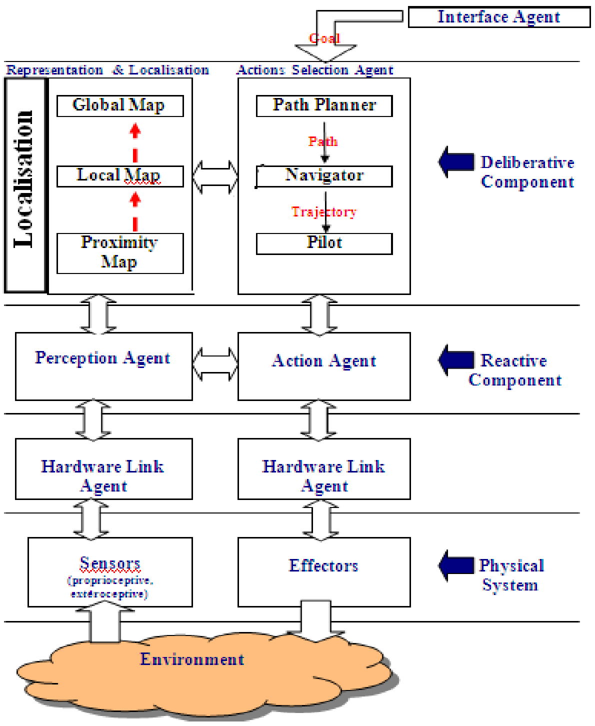

Fig 5: EAAS Architecture

EAAS architecture consists in five agents: interface agent, actions selection agent, perception agent, action agent and hardware link agent. The interface agent is the high level of our control architecture. It must generate a succession of goal, or missions for the actions selection agent, according to the general mission of the robot. It is the “ultimate” robot autonomy concept: the robot generates itself its own attitudes and its own actions by using its own decisions. The perception agent manages the processing of incoming data (the sensor measurements) and creates representations of the environment. The actions selection agent must choose the robot behavior according to all information available and necessary to this choice: the fixed goal, representations and the robot localization. The actions selection agent contains a path planner, a navigator and a pilot. The path planner may take a goal as input and give a path for achieving the goal as output. The Navigator must translate a path into a trajectory for the pilot. The path does not take into account physical constraints of the robot, but the trajectory that it delivers must integrate them. The function of the pilot is to convert this trajectory into orders to be performed by the action agent. The action agent consists of a set of behaviors controlling the robot effectors. The hardware link agent is an interface between the software architecture and real robot. Changing the real robot require the use of a specific agent but no change in the overall architecture.

5. REMOTE CONTROLE SOFTWARE ARCHITECTURE

A Software architecture has been defined to make remote control of mobile Robot possible [25] [26]. Our software architecture is based on a set of independent agents running in parallel. On the left side of figure 6, the server side is represented. It is basically composed of three main agents: “Connection Manager” which manages the different connected clients according to a Control Algorithm. This one is chosen by the designer of the system depending on the application: master/slave, priority, timeout... The “Media” agent communicates with the camera in order to broadcast signals (video, images) of the mobile robot in its environment. The “SMA EAAS” (EAS Architecture for Autonomous Systems) which represents our control architecture. EAAS architecture is a hybrid control architecture including a deliberative part (Actions Selection Agent) and a reactive part. The reactive part is based on direct link between the sensors (Perception Agent) and the effectors (Action Agent).

The right side of the figure represents the client side. Agents are loaded in a web navigator. The “Remote Client” corresponds to a graphical user interface which allows the user to send orders to the mobile robot and receive information about the environment. “Sender” and “Receiver” agents are used to allow the communication between the client and the server. “Pinger” and “Ponger” agents are used to observe dynamically the network.

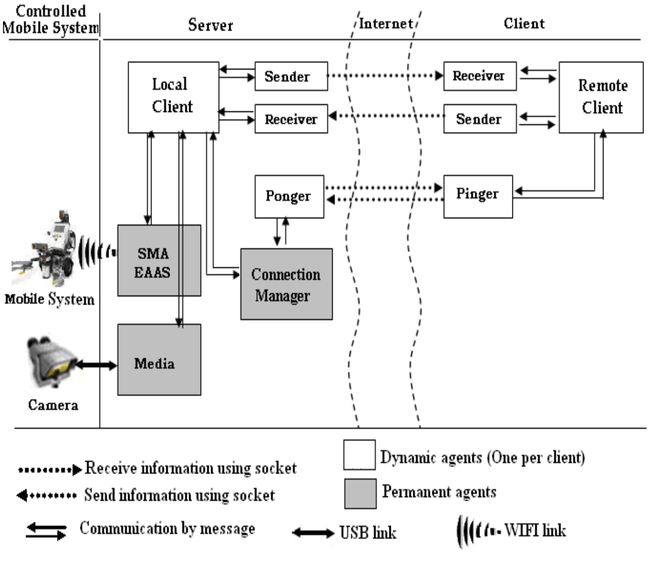

Fig 6: Remote control software architecture proposed

If the connexion is accepted, the “Connection Manager” will inform the “Local Client” agent which achieves the interface with the “SMA EAAS” to transmit orders transmission to the mobile robot.

6. REALISATION

During the project development, different configurations were tested in different environments. The aim is to develop a more reliable system architecture that can be used in the real world. Our Khepera mobile robot was controlled to explore the laboratory we are working in. In another test, the Lego mobile robot was controlled to push a ball to the goal while avoiding several static obstacles.

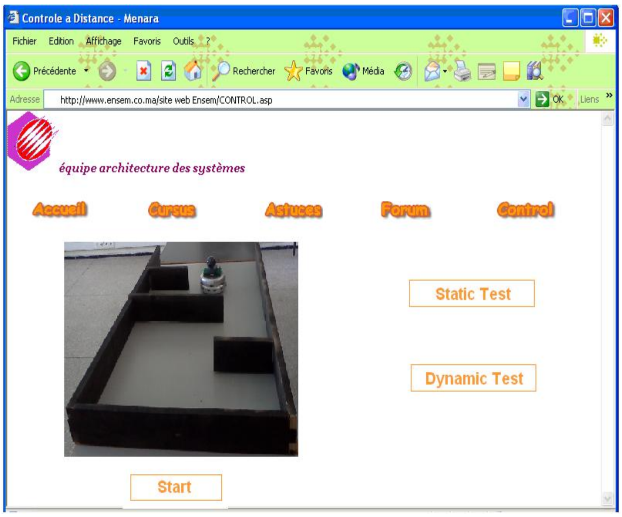

Fig 7: The Control Interface

The user interface (Figure 7) is designed with the intention of making it easy for researchers and students to interact with the mobile robot. A simple interface is designed to provide as much information as possible for remote control. This web interface consists of several Java Applets as shown in figure 7. It can work on any web browser that supports Java1.2 or above. The user can directly control the mobile robot by clicking the start button on the control panel. The image display applet shows the visual feedback in a continuous jpeg image. The forum service allows users to send messages to each other, private or broadcast in order to interchange their ideas over the remote control subject. The remote user is invited to test the connection using the statistical or the dynamical way, before or during taking the control by clicking on the buttons labelled statistic or dynamic test respectively. This user interface allows students to undergo a distance learning with the opportunity to test their ability on line.

7. CONCLUSION

In this paper, we have presented a Web-based remote control application so that Internet users, especially researchers and students, can control the mobile robot to explore a dynamic environment remotely from their home and share this unique robotic system with us.

The long-term goal of our research is towards real-world applications such as tele-teaching, tele-maintenance, tele-expertise and tele- production.

In future work we hope to increase the intelligence of the agents in our control architecture to provide a telerobot with a high degree of local intelligence to handle restricted bandwidth and transmission delay of the network and to integrate multiple mobile robots into a telerobotics system to achieve redundancy and robustness. This will pave the way for the remote exploration of an unknown and complex environment through the Internet.

8. REFERENCES

[1] Le parc, P., Ogor, P., Vareille, J., Marce, L. "*Web based remote control of the mechanical systems"*, Proceedings of the IEEE International Conference on Software Telecommunications and Computer Networks, Split, Croatie (Year of Publication: 2002).

[2] Sayouti, A., Qrichi Aniba, F., Medromi. "Tele-robotic over Internet Based on Multi-agents System". Proceedings of International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES'2008), Dalian, China, July 27-31, 2008.

[3] Goldberg, K., Siegwart, R. "*Beyond Webcams : an introduction to online robots"* (The MIT Press, 2001).

[4] Goldberg, K., Gentner, S., Sutter, S., Wiegley, J. "The mercury project: a feasibility study for internet robotics", *IEEE Robotics & Automantion Magazine*, pp. 35-40, 2000.

[5] Taylor, K., Dalton, B. "*Issues in internet telerobotics"*, Proceedings of the FSR'97 International Conference on Field and Service Robots (Year of Publication: 1997).

[6] Schulz, D., Burgard, W., Cremers, A. B. "*Predictive simulation of autonomous robots for tele-operation systems using the world wide web"*, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System, Victoria, B.C., Canada (Year of Publication: 1998).

[7] Simmons, R., *Xavier : "An autonomous mobile robot on the web",* Proceedings of the International Workshop On Intelligent Robots and Systems (IROS), Victoria, Canada (Year of Publication: 1998).

[8] Stein, M. R." *Painting on the world wide web: the pumapaint project"*, Proceedings of the IEEE IROS'98 Workshop on Robots on the Web, pages Victoria, Canada (Year of Publication: 1998).

[9] Saucy, P., Mondada, F. Khepontheweb: "Open access to a mobile robot on the internet", *IEEE robotics and automation magazine*, pp. 41-47, March 2000.

[10] Kosuge, K., Kikuchi, J., Takeo, K. *VISIT: "A Teleoperation system via computer Network"*, Proceedings of the IROS'98: Workshop on Web Robots, Victoria, Canada (Page: 61-66 Year of Publication: 1998).

[11] Hu, H., YU, L,. Tsui, Zhou, P. "Internet-based Robotic Systems for Teleoperation", International *Journal of Assembly Automation*, Vol. 21, n. 2.

[12] Le Parc, P., Vareille J. Marcé, "*Long distance remote control over Internet: a reliability challenge"*, Proceedings of the XVI Workshop on Supervising and Diagnostic of Machining Systems., Karpacs, Poland (Year of Publication: 2005).

[13] Sayouti, A., Qrichi Aniba, F., Medromi, H., Lebbat, A. "*Remote Control Based on Multi-agents Systems"*, Proceedings of the Second IEEE International Conference on Research Challenges in Information Science (RCIS), Marrakech, Morocco (Year of Publication: 2008).

[14] Novales, C., Mourioux, G., Poisson, G. "*A multi-level architecture controlling robots from autonomy to teleoperation"*, Proceedings of the First National Workshop on Control Architectures of Robots. Montpellier (Year of Publication: 2006).

[15] Brooks, R.A."A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation"*, vol. 2, n. 1, pp. 14–23, 1986.

[16] Albus, J. S. "*4D/RCS A Reference Model Architecture for Intelligent Unmanned Ground Vehicules"*, Proceedings of the SPIE 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando (Year of Publication: 2002).

[17] Volpe, R. "CLARAty: Coupled Layer Architecture for Robotic Autonomy" (JPL Technical Report D-19975, 2000).

[18] Alami, R., Chatila, R., Fleury, S., Ghallab, M., F. Ingrand, "An architecture for autonomy". *The International Journal of Robotics Research, Special Issue on Integrated Architectures for Robot Control and Programming*, vol. 17, n. 4, pp. 315-337, 1998.

[19] Dalgalarrondo, A. « Intégration de la fonction perception dans une architecture de contrôle de robot mobile autonome », Ph.D. Thesis, Université Paris-Sud, Orsay, France, 2001.

[20] Rosenblatt,J. *DAMN: "A Distributed Architecture for Mobile Navigation",* Proceedings of the 1995 AAAI Spring Symposium on Lessons Learned from Implemented Software Architectures for Physical Agents, AAAI Press, Menlo Park, CA (Year of Publication: 1995).

[21] Sayouti, A., Qrichi Aniba, F., Medromi, H., Radoui, M. "*Tele-robotic over Internet Based on Multi-agents System"*, Proceedings of the Seventh International Conference on Distributed Computing and Applications for Business, Engineering and Sciences (DCABES), Dalian, China (Year of Publication: 2008).

[22] Ferber, J. "Les systèmes multi-agents", vers une intelligence collective (Paris, InterEditions, 1995).

[23] Drogou, A., Ferber, J. "Multi-agent simulation as tool for modelling societies: Application to social differentiation in ant colonies", Proceedings of the MAAMAW'92, Viterbo (Year of Publication: 1992).

[24] Ferber, J. "Multi-agents systems, an introduction to distributed artificial intelligence" (Addison-Wesley, 1999).

[25] Sayouti, A. "Conception et Réalisation d'une Architecture de Contrôle à Distance Via Internet à Base des Systèmes Multi-Agents". Phd. Thesis, ENSEM, Hassan II University, 2009.

[26] Sayouti, A., Medromi, H. "Chapter Title: Autonomous and Intelligent Mobile Systems based on Multi-Agent Systems, Book Title: Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications", INTECH, http://www.intechweb.org, 2011.