4th International Workshop on Practical Applications of Agents and Multiagent Systems

IWPAAMS 2005

20th to 21st of October

UNIVERSIDAD DE LEON

4th International Workshop on Practical Applications of Agents and Multiagent Systems - IWPAAMS 2005

Proceedings IWPAAMS 2005

Acknowledgements

The IWPAAMS'2004 is organized in collaboration with:

- UNIVERSIDAD DE LEÓN
- DIPUTACIÓN PROVINCIAL DE LEÓN
- JUNTA DE CASTILLA Y LEÓN
- MINISTERIO DE EDUCACIÓN Y CIENCIA
- CAJA ESPAÑA
- TELEFÓNICA I + D

The organizing committee of IWPAAMS'05 would like to thank them for their appreciated contribution to the success of the conference.

© Universidad de León

Secretariado de Publicaciones

© Los autores

ISBN: 84-9773-222-7
Depósito Legal: LE-1867-2005
Impresión: Universidad de León. Servicio de Imprenta

TABLE OF

Towards a Multi-Agent Software Architecture for Human-Robot Integration

Blanco J.L, Cruz A., Galindo C., Fernández-Madrigal J.A., and González J.

E.T.S.I. Informática – Universidad de Málaga
Campus Teatinos, 29071, Málaga, Spain
jlblanco@ctima.uma.es
http://www.isa.uma.es

Abstract. A recent and promising area of research in robotics is aimed to develop robots that share the same physical environment as humans. In this area, the most relevant issue is not autonomy of the robot but the intelligent interaction or cooperation with humans. Some particular instances are service robots (museum guides, surveillance, etc.) and assistant robots (elderly care, telesurgery, etc.), where the latter usually exhibit closer interaction with humans than the former. In this paper we identify and use multi-agent technology as a suitable approach for developing assistant robot architectures, since it enables a very close interaction –namely, *integration*- with humans (by considering the human as part of the robot architecture) and it is also capable of learning the best way of coordinating both robot and human skills (through reinforcement learning). We illustrate the approach with our robotic wheelchair SENA, a mobile robot intended for assistance to handicapped persons.

1 Introduction

Robotic applications that interact with humans have recently attracted great attention from the scientific community. One of the reasons is that the presence of a human that can interact with the robot may relax some of the requirements demanded from complete autonomous robots.

Among robots that interact with humans, maybe the most studied ones are *service robots* (that carry out limited tasks in human populated scenarios such as museums, hospitals, etc.) and *assistant robots* (which can not work without humans at all, for example surgery robots, assistant robots for elderly people, artificial limbs, etc). Human-robot interaction becomes an important requirement in these applications. They also bear two important characteristics that make them distinctive from others: on the one hand, they impose some critical issues such as operating robustness, physical safety, and human-friendly interaction (which have been coped recently by the *Human Centered Robotics* (HCR) paradigm ([11], [13]). On the other hand, the human becomes in fact so integrated into the assistant robot that it is no more merely a part of the environment.

In the literature, human participation in the robotic operation is not a new idea. Some terms have been coined to reflect this, such as *cooperation* [19], *collaboration* [5], or *supervision* ([5], [15]), but all of them deal with the human as being external to the robot. We propose here to take a step further: to consider the human as one of the components of the robotic application. This is specially evident in applications where the human can be considered *physically* as a part of the robot, such as robotic wheelchairs or artificial limbs. In these situations we should talk of *human-robot integration* rather than *human-robot interaction.* We can obtain the following benefits from such integration:

-Augment the robot capabilities with new skills. Humans can perform actions beyond the robot capabilities, i.e. to open a door, to take an elevator.

-Improve the robot skills. Humans may carry out actions initially assigned to the robot in different and sometimes more dependable ways, or may complete robot actions that have failed.

Our approach uses a multi-agent software robotic architecture, that we call MA^2CHRIN (the acronym for "Multi-Agent Architecture for Close Human-Robot Integration") for such integration. This framework is more appropriate than a classical hybrid layered (deliberative-reactive) architecture [16], since a software agent is the closest we can be to represent the human within the architecture (as a part of it) under the current software technology.

The work presented in this paper is a first step towards a completely integrated human-robot system. At this point, the human is not yet an agent, but his/her capabilities can be distributed inside any agent that needs them for augmenting its skills. Our design has several immediate advantages concerning efficiency in human-robot coordination and at the same time enables a high degree of integration of the human into the architecture. In a future work we will analyze the advantages of considering the human as another, complete agent that communicates with the rest.

The following sections of the paper describe our framework in more detail. Section 2 gives an overview of the proposed multi-agent architecture. The semantics that enable inter-agent communication and intra-agent mental states is described in Section 3. Section 4 is devoted to the learning process that allows each agent to select which internal skill is most appropriate to solve a given action (which includes human skills). Section 5 illustrates our approach with a real assistant robotic application. Finally some conclusions and future work are outlined.

2 Architecture Overview

MA^2CHRIN, which is an evolution of a non-agent-based architecture presented elsewhere [9], is a Multi Agent System (MAS) composed of a variety of agents. Its main feature is that human skills are integrated into those agents. For instance, the human can help the robot to perceive world information (as we have demonstrated previously in [4]), or to provide a plan for achieving a goal (interacting with the planner agent, as we have shown in [8]), or even to physically perform actions like open a door or call an elevator.

This is supported through the use of the *Common Agent Structure (CAS)* as the skeleton of each agent (see fig. 1). The abilities of the agent, both human and robotic, are the so-called *skill units* (see fig.2). *Robotic skill units* represent abilities implemented by software algorithms, while *human skill units* permit the connection between humans and architecture agents, enabling them to perform human abilities (i.e., open a door) through appropriate interfaces (i.e., voice communication).

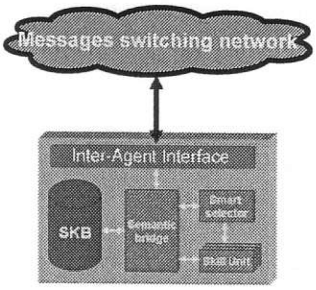

Fig. 1. The common agent structure (CAS). All the software agents are designed using this structure. The number of skill units contained into each agent is variable and it depends on its functionality as well as on the human and robot capabilities that the CAS provides. See text for more detail.

Among the different possibilities for accomplishing its tasks an agent should be able to learn over time the most convenient one given its past experience and the current conditions of the environment [14]. Within the CAS skeleton, the *Smart Selector* is in charge of the agent's learning process, and it is active every time the agent operates, in order to adapt and optimize the agent performance under changeable environmental conditions, as commented in section 4.

All skill units considered by the *Smart Selector* to accomplish a requested action pursuit the same goal (i.e. to reach to a destination in the case of navigating), but they may exhibit differences in the way they are invoked, especially in the case of human units.

The translation from a generic action requested by an external agent into the skill unit parameters is carried out by the *Semantic Bridge*. This component also permits the internals of the agent (Smart Selector and Skill Units) to request/provide data to other agents of the architecture.

The *Semantic Bridge* is also in charge of maintaining the *Semantic Knowledge Base* (SKB) that represents the internal mental state of the agent in terms of intentions and beliefs. Such an internal mental state is updated by the *Semantic Bridge* with incoming messages or requests from other agents. Communications between agents relies on the *Inter-Agent Interface* that provides communication primitives to send/receive messages following a fixed semantics as commented in more detail in section 3.

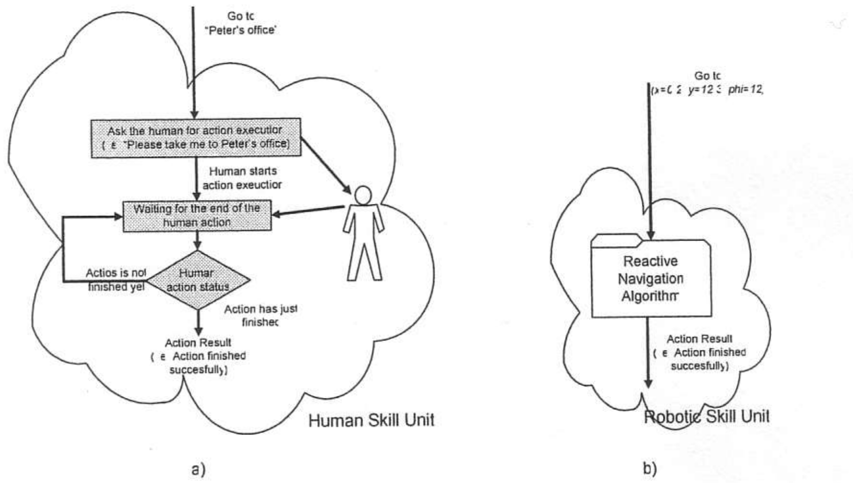

Fig. 2. Internal scheme of two skill units. a) A human skill unit that permits a person to manually guide the robot through a dialogue using appropriate interfaces, i.e. via voice. b) a robotic skill unit that also performs navigation.

3 A Semantics for Agents

In contrast with traditional software architectures, where communication between different pieces of software usually makes intensive use of the client-server paradigm (just using procedural calls, RPC or any other function invocation mechanism), MAS imposes a much richer set of interactions between agents. Thus, if communications are expected to reflect and affect the agents' internal mental states, they should be more than mere raw data, but *information* about agent attitudes. Hence, the semantics of interactions must be well defined.

Communications in MA^{2}CHRIN are currently based on the most commonly used scheme for MAS, which is *message passing*. In this approach we use a well-defined, standardized message format, the one proposed by FIPA, namely ACL [7]. Moreover, we have been highly inspired by the CAL specification [6] in order to define the underlying semantics of communicative acts.

Agent mental attitudes in our architecture are defined using these operators:

- *Beliefs*: the operator $B_i\,p$ means that agent i beliefs that fact p holds true.
- *Intentions*: the operator $I_i\ a$ means that agent i intents a to hold true. Semantic meaning of an intention [1] differs from the concept of *goal* usually used in robotics. Intentions are usually used for invoking the execution of actions.

The set of communicative acts or *performatives* that we are using in MA^{2}CHRIN, and their associate feasible preconditions (FP) and rational effects (RE), include:

- **Inform**: The intention of an agent i of letting another agent j to know something that i currently beliefs. Its formal model results as: $\{<i, \text{inform}\ (j,\phi)>;$ FP: $B_i\ \phi \wedge \neg B_i\ B_j\ \phi;$ RE: $B_j\ \phi\}$

- **Request**: Sender agent requests the receiver to execute a given action. Its formal model is: {<*i*, request (*j*, *a*)>; FP:FP(*a*) [*i*/*j*] $\wedge$ B_i Agent(*j*, *a*) $\wedge$ $\neg B_i$ I_j Done(*a*); RE: Done (*a*)}, where we have used the operators: *Agent(j,a)*, meaning that agent *j* is able of performing action *a*; *Done(a)* which means action *a* has to be executed; and *FP(a)* [*i*/*j*] representing the part of FP(*a*) which are mental attitudes of *i*.[1]
- **Agree**: The agreement of an agent *i* to perform a requested action for some other agent *j*: {<*i*, agree(*j*, *a*)> ≡ <*i*, inform (*j*, I_i Done(*a*))>; FP: B_i α $\wedge$ $\neg B_i$ B_j α; RE: B_j α}, where for clarity, $\alpha = I_i$ Done(*a*).
- **Cancel**: Informs to an agent *j* that agent *i* has no longer the intention of agent *j* performing the action *a*: {<*i*, cancel (*j*, *a*)> ≡ <i, inform(j, $\neg I_i$ Done(*a*)>; FP: $\neg I_i$ Done(*a*) $\wedge$ B_i B_j I_i Done(*a*); RE: B_j $\neg I_i$ Done(*a*)}
- **Failure**: Agent *i* is intended to perform action *a* while its preconditions are fulfilled, but it was not possible to complete it and currently it intents no longer to try it: {<*i*, failure(*j*, *a*)> ≡ <*i*, inform(*j*, Done(*e*,Feasible(*a*) $\wedge$ I_i Done(*a*)) $\wedge$ $\neg$Done(*a*) $\wedge$ $\neg I_i$ Done(*a*))>; FP: B_i α $\wedge$ $\neg B_i$ B_j α ; RE: B_j α}, where for clarity, α= Done(*e*, Feasible(*a*) $\wedge$ I_i Done(*a*)) $\wedge$ $\neg$Done(*a*) $\wedge$ $\neg I_i$ Done(*a*), and the operator Feasible(*a*) means that preconditions for action *a* are met at that time.
- **Refuse**: Agent *i* refuses the request of agent *j* for performing *a,* where ϕ is the reason of rejection, or true if reason is not supplied. Another possibility is ϕ being a textual message in natural language. Its model is: {<*i*, refuse(*j*, *a* ,ϕ)> ≡ <*i*, inform(*j*, $\neg$Feasible(*a*)>; <*i*, inform(*j*, ϕ $\wedge$ $\neg$Done(*a*) $\wedge$ $\neg I_i$ Done(*a*)>; FP: B_i $\neg$Feasible(*a*) $\wedge$ B_i B_j Feasible(*a*) $\wedge$ B_i α $\wedge$ $\neg B_i$ B_j α; RE: B_j $\neg$Feasible(*a*) $\wedge$ B_j α}, where a_1 ; a_2 means that both actions are sequenced, and for clarity $\alpha = \phi \wedge$ $\neg$Done(*a*) $\wedge$ $\neg I_i$ Done(*a*).
- **Inform-ref**: This performative contains a β expression, whose meaning is not standardized, but it is assumed that the receiver agent should be able to evaluate it

[1] Note that FP includes a precondition stating that the requesting agent doesn't belief that agent *j* already has the intention of performing *a*, a condition that becomes false after receiving an *agree* performative.

through the proper translation in its semantic bridge. Its formal model is: {<i, inform-ref(j, β)> ≡ <*i*, inform(j, Eval(β))>; FP: B_i I_j Done(<i, inform-ref(j, β)>); RE: B_j Eval(β)}, where Eval(β) represents the expression β evaluated at agent *j*.

Please note that while we have made many simplifying assumptions in order to reduce the complexity of standard CAL, our semantics remains very close to the one proposed there and it is self-consistent. In order to transport the messages between agents, the performatives are sent inside an ACL formatted string, which is coded into an ACL compliant message in plain text format.

4 Learning Intra-Agent Skill Selection

In our architecture, agents must learn how to select the appropriate skill unit (either human or robotic) in order to carry out the agent actions. We have chosen for that purpose *Reinforcement Learning (RL)* [10] to be implemented in our Smart Selector, since RL has a thoroughly studied formal foundation and it also seems to obtain good results in mobile robotics tasks ([12], [17]).

In short, in RL an agent in a state *s* executes some action *a* turning its state into *s'* and getting a reinforcement signal or reward *r*. Those experience tuples (*s,a,s',r*) are used for finding a policy π that maximizes some long-run measure of reward.

There is a comprehensive bunch of methodologies for modelling and solving RL problems. A well-known solution are *Markov Decision Processes* (*MDPs*), that can be defined by: a set of states *S*, a set of actions *A*, a reward or reinforcement function $R : S \times A \rightarrow \Re$, and a state transition function $T : S \times A \rightarrow \Pi(s)$, where $\Pi(s)$ is a probability distribution over the set *S*.

Upon these sets and functions, an optimal policy π that maximizes the obtained rewards can be computed through the definition of the *optimal value* of a state $V^*(s)$, which is the expected reward that the agent will gain if it starts in that state and executes the optimal policy:

$$V^*(s) = \max_a (R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V^*(s')), \forall s \in S \cdot \quad (5$$

Where parameter γ is a *discount factor* that represents how much attention is paid to future rewards. The optimal policy is then specified by:

$$\pi^*(s) = \arg\max_a (R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V^*(s')) \cdot \quad (2$$

Both reinforcement and state transition functions are named a *model*. However, such a model is not always known in advance; in fact, most of robotics applications cannot provide that prior knowledge. For model-free problems as the one at hand, we can use a widely known approach called *Q-learning*. It uses the following optimal value function Q^* instead of $V^*(s)$, which can be recursively computed on line by means of:

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)) \cdot \quad (3)$$

Parameter α is the *learning rate*, and it must slowly decrease in order to guarantee convergence of function Q to Q^*. Once the optimal Q-function Q^* is obtained, the optimal policy can be fixed as it was stated in (2).

For illustrating Q-learning in our Smart Selectors, we focus our attention on the *Navigation Agent*. This agent is in charge of the performing motion actions of the robot (in particular, of our robotic wheelchair). There are three skill units available in this agent: a human skill unit that puts human in charge of movement through a joystick, a path tracker algorithm that follows a previously calculated path, and a robotic reactive skill unit that implements a reactive navigation algorithm [1].

The mathematical formulation of the Q-learning technique for the *Navigation Agent* is:

- S={SUCCESS, FAILURE, NAV_OK, NAV_NOK}. SUCCESS means that the robot has reached its objective. FAILURE represents those situations where the vehicle stops outside its goal, due to some unexpected problem. If the robot is travelling with normality, it is in NAV_OK state. However, if difficulties arise during navigation, the robot turns to NAV_NOK state. Transitions among states are displayed in fig. 4.
- *A* simply matches the set of skill units.
- The reinforcement function *R* is considered as a sum of relevant factors, namely: ability for obstacle avoidance, energetic consumption, path tracking curvature changes, and distance to the goal.

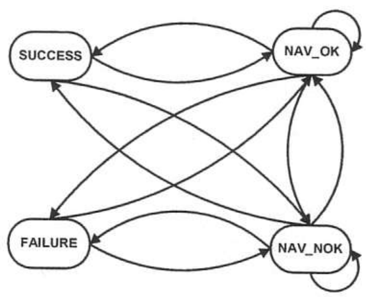

Fig. 4. State transition diagram for the Navigation Agent.

5 A Real Robotic Assistant Application

MA^{2}CHRIN has been tested on an assistant robot called SENA (see fig 5). It is a robotic wheelchair based on a commercial powered wheelchair that has been equipped with several sensors and an onboard computer to reliably perform high-level tasks in indoor environments. SENA accounts for Wi-Fi connection capabilities that improve to a great extend the possibilities of our robot, ranging from tele-operation of the vehicle to the human driver access to internet.

Our tests have been carried out within our lab and near corridors, in which the user can select a destination via voice. Since the main operation of SENA is navigation, we have focussed our experiences on the learning process of the agent devoted to this type of task. We have considered three different skill units within the navigation agent, as described in section 4.

For a navigation task, the Planner agent interacts with the user (as described in [8]) for constructing the best route to the goal. With the route available, the Navigation Agent executes each navigation step. All the communications that arise in this scheme follow the semantics described in section 3. When each navigation step arrives at the Navigation Agent, its Semantic Bridge translates it to the Smart Selector for performing the action. The Smart Selector then runs the Q-learning algorithm.

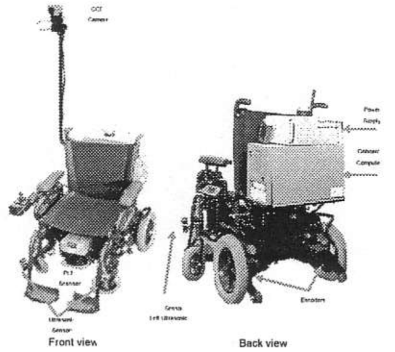

Fig. 5. Two views of the SENA robotic wheelchair in which we have evaluated our human-robot integration architecture.

Our findings indicate that the Q-function really converges to a certain value after an appropriate number of learning steps. Fig. 6 shows, for states FAILURE, NAV_OK and NAV_NOK, how Q-function of every action (HUMAN, ROBOT_DEL and HUMAN, respectively) finally converges.

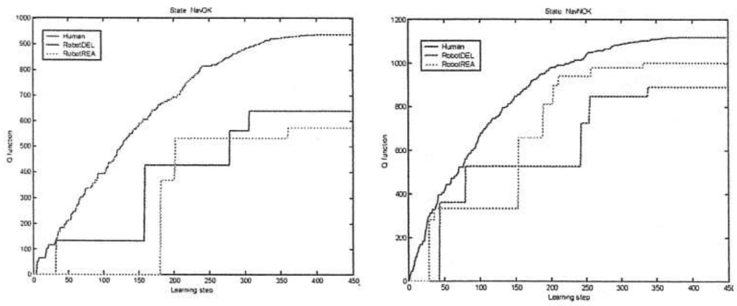

Fig. 6. Q function values for states NAV_OK and NAV_NOK. For each state, the best choices are Robotic Deliberative and Human, respectively.

It can also be noticed that, for a given state, the maximum value of the Q function changes among different actions. This means that, though initially Q-learning chooses a certain action, as execution progresses another action can be discovered as the best suited option for navigation. To illustrate this, refer to fig. 7: until learning step 10, it is ROBOT_DEL who provides best Q function value; then, until approximately learning step 70, best action turns to ROBOT_REA; finally, HUMAN becomes the most fitting value.

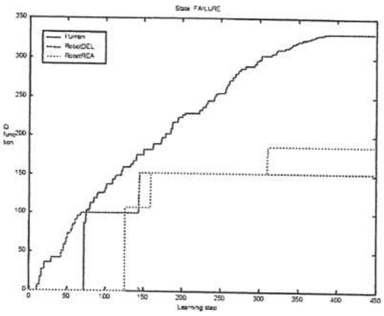

Fig. 7. Q function values for the FAILURE state. In this case the best choice is Human.

Finally, we have also noticed that the human skill unit is mostly learned for those states corresponding to risky or difficult situations (FAILURE and NAV_NOK states).

6 Conclusions and Future Work

Usually, human-robot interaction is considered to be as a simple communication between the robot architecture and the human. However, we claim that in assistant robotics, a much stronger interaction is needed: the human, in fact, must be *integrated into the robotic architecture*.

For that purpose, we have presented in this paper a multi-agent robotic architecture that enables such human-robot integration. We believe that a multi-agent system is more appropriate than conventional approaches since agents are closer to represent human as a part of the system than other software constructions (modules, procedures, etc.). Agents, as well as humans, have intentions, mental states, and use some semantics in their communications. In addition, they posses some learning capabilities that allow them to adapt to environmental changes and to achieve good performances over time.

This paper is a first step towards the inclusion of the human as an agent within the architecture. At this stage, the human is not yet an agent, but his/her capabilities are distributed inside any agent that needs them for augmenting its skills.

When an agent is endowed with a set of skill units for performing some action, it must decide which skill unit (including both robotic and human ones) is the best at each situation. We have implemented a Q-learning procedure that learns this association over time, trying to optimize the behavior of the system at long-term.

We have also defined the semantics for the agents to communicate and maintain their internal mental states, as well as how this semantics is translated into practical requests for the algorithms of the architecture.

In the future, we will analyze the effects of including the human as one of the agents of the architecture. We will also continuing on working with assistant robots (such as our robotic wheelchair) in order to implement solutions for the real requirements of assistance applications.

References

1. Blanco J.L, Gonzalez J., Fernandez-Madrigal J.A. *The PT-Space: A new Space Representation for Non-Holonomic Mobile Robot reactive Navigation*. Tech. Report, University of Malaga, 2005.
2. Cohen P.R. and Levesque H.J. *Intention is choice with commitment*. Art. Intellig., vol. 42, n. 3, 1990.
3. Fernandez-Madrigal J.A. and Gonzalez J. *Multi-Hierarchical Representation of Large-Scale Space*. International Series on Microprocessor-based and Intelligent Systems Engineering. Vol. 24, Kluwer Academic Publishers, Netherlands, 2001.
4. Fernandez-Madrigal J.A., Galindo C., and Gonzalez J. *Assistive Navigation of a Robotic Wheelchair using a Multihierarchical Model of the Environment*, Integrated Computer-Aided Engineering, vol 11., pp. 309-322, 2004.
5. Fong T.W. and Thorpe C. *Robot as Partner: Vehicle Teleoperation with Collaborative Control,* Proc. of the NRL Workshop on Multi-Robot Systems, 2002.
6. Foundation for Intelligent Physical Agents. ACL Communicative Act Library Specification, 2002.
7. Foundation for Intelligent Physical Agents. ACL Message Structure Spec., 2002.
8. Galindo C., Fernandez-Madrigal J.A., and Gonzalez J.. *Hierarchical Task Planning through World Abstraction*. IEEE Trans. on Robotics, 20(4), pp. 667-690, 2004.
9. Galindo C., Gonzalez J., and Fernandez-Madrigal J.A. *An Architecture for Close Human-Robot Interaction. Application to Rehabilitation Robotics*, IEEE Int. Conf. on Mechatronics and Automation (ICMA'2005), Ontario (Canada), July-Agust 2005.
10. Kaelbling L.P., Littman M.L., Moore A.W. *Reinforcement Learning: A Survey*. Journal of Artificial Intelligence Research 4 (1996) 237–277.
11. Khatib O., *Human-Centered Robotics and Haptic Interaction: From Assistance to Surgery, the Emerging Applications*. 3rd Int. Workshop on Robot Motion and Control, 2002.
12. Liu J., Wu J. *Multi-Agent Robotic Systems*. CRC Press (2001)
13. Morioka K., Lee J.H., and Hashimoto H. *Human Centered Robotics in Intelligent Space* Proc. of the IEEE ICRA'02, Washington, DC.
14. Murch R. and Johnson R. *Intelligent software agents*. Prentice Hall 1999.
15. Scholtz J. *Theory and evaluation of human-robot interaction*, Proc. of the 36th International Conference on System Sciences, Hawai, 2003.
16. Simmons R., Goodwin R., Haigh K., Koening S., and Sullivan J. *A Layered Architecture for Office Delivery Robots*. 1st Int. Conf. on Autonomous Agents, pp. 235-242, 1997.
17. Smart W.D., Kaelbling L.P. *Effective Reinforcement Learning for Mobile Robots*. International Conference on Robotics and Automation (2002)
18. Sutton R.S., Barto A.G. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA (1998).
19. Tahboub K.A. *A Semi-Autonomous Reactive Control Architecture*, J. Intell. Robotics Syst., vol. 32, n. 4, pp. 445-459, 2001.