Multi-agent Architecture Model for Driving Mobile Manipulator Robots

A. Hentout, B. Bouzouia and Z. Toukal
Division Productique et Robotique, Centre de Développement des Technologies Avancées (CDTA), Algeria
Corresponding author E-mail: ahentout@cdta.dz, hentout_abdelfetah@hotmail.com

Abstract: *In this article, we present generic hierarchical behavior-based architecture model for driving mobile manipulator robots. Two behaviors are of high-level. They constitute the Supervisory agent, which manages the global system. Two others are of intermediate-level and finally one behavior is of low-level. These last ones constitute the Mobile Robot agent and the Manipulator Robot agent controlling, respectively, the mobile base and the manipulator arm. The choice of the suggested model is justified by the generic character of the proposed agent model and by the possibility of integrating the whole in a distributed robotic system. The model is formalized in Agent UML from the conceptual level to the implementation level. The interaction between the various agents is modeled by the use of the interaction diagrams of Agent UML (states and protocol diagrams).*
Keywords: *Driving Architecture, Behavior-based Architecture, Multi-agent System, Mobile Manipulator Robot, RobuTER, Agent UML.*

1. Introduction

The development of the robotic systems constitutes a very active field of research. During this last decade, we note the emergence of a new research discipline: The multi-agent systems. These latter ones are largely adopted in complex systems and distributed applications and, in particular, those dedicated to drive robotic systems as mobile robots, flexible cells, manipulator robots and mobile manipulator robots.
Multi-agent control is necessary when more than one robot is used to execute tasks, when a robot must coordinate the use of its own resources (coordinating the arm and the platform for a mobile manipulator robot) or when a robots society function independently on multiple tasks in a shared environment. The multi-agent systems are very suitable as regards the new software requirements thanks to their properties of decentralization, modularity, autonomy, effectiveness, reliability and reutilisability of agents for the implementation of other systems (Bonasso R. P., Firby R. J., Gat E., Kortenkamp D., Miller D., Slack M., 1997). Nevertheless, the development of this type of systems includes several problems. The absence of methodologies of design and implementation covering all the life cycle of the multi-agent system and allowing to understand, to represent, to analyze, to conceive and to implement such systems makes this task very difficult. We can also add other problems as the choice of agents' types, communication protocol, communication and interaction mode, conflicts management, environment modeling, tasks planning, scheduling, etc.

There is a large body of literature describing distributed architectures for driving robotics. In the following, we shall examine several of the better known robot architectures.
We can find a survey of existing systems composed of multiple autonomous mobile robots exhibiting cooperative behavior in (Cao Y. U., Fukunaga A. S., Kahng A. B., Meng F., 1995), (MacKenzie D. Ch., 1996) and (Cao Y. U., Fukunaga A. S., Kahng A. B., 1997). In addition, a state of the art in distributed mobile system is presented in (Parker L. E, 2000), and in particular, architectures that have been demonstrated in physical robot implementation. Balch in (Balch T., Arkin R. C., 1994) studied the importance of communication in robotic societies and described in (Balch T., Boone G., Collins T., Forbes H., MacKenzie D., Santamaria J.-C., 1997) the design and implementation of three eactive trash-collecting robots (Io, Ganymede and Callisto) including details of multi-agent cooperation, temporal sequencing of behaviors for task completion. Agah and *al*. in (Agah A., Bekey G.A., 1995) treated the problem of mobile robot colonies with populations of up to 100 robots. Robots in the colony perform tasks in dynamics world including different types of obstacles. They perform tasks including gathering of small objects, decomposing of large objects to smaller ones, and collecting of large, non-decomposable objects that require the cooperation of two robots. When a robot encounters such a task, it sends a call for help to the rest of the colony, and one or more other robots will respond.
Innocenti and *al*. (Innocenti B., Lopez B., Salvi J., 2003) developed a multi-agent architecture comprising the following agents: *Task Planning agent* which provides