COMMUNICATION AT ONTOLOGICAL LEVEL IN COOPERATIVE MOBILE ROBOTS SYSTEM

Lucia Vacariu, Mihai Chintoanu
Department of Computer Science, Technical University of Cluj Napoca, 26 Baritiu str, Cluj Napoca, Romania
Navigon Company Cluj Napoca Subsidiary, Cluj Napoca, Romania

Gheorghe Lazea, Octavian Cret
Department of Automation, Technical University of Cluj Napoca, Cluj Napoca, Romania
Department of Computer Science, Technical University of Cluj Napoca, Cluj Napoca, Romania

Keywords: Mobile robots, Multiagents system, Communication, Ontology, Cooperation.

Abstract: Mobile robot applications are faced with the problem of communicating large amounts of information, whose structure and significance changes continuously. A traditional, layered style of communication creates a cooperation problem as fix protocols and message meaning cannot mediate the dynamic, novel types of behaviors mobile robots are acquiring in their environment. We propose by contrast, a non-hierarchical communication control mechanism based on the software paradigm of multiagent systems that have specialized ontologies. It is a communication at ontological level, which allows efficient changes in the content of the messages among robots and a better adaptability and specialization to changes. The intended application is a cooperative mobile robot system for monitoring, manipulating and cleaning in a supermarket. The focus of the paper is to simulate a number of well-defined, controllable and repeatable critical ontological situations encountered in this environment that validate the system cooperation tasks.

1 INTRODUCTION

Research in mobile robotics has achieved a level of maturity that allows applications to move from laboratory into public-level applications. This leads necessarily to a situation where several robots, perhaps with different goals and different internal representations, must cooperate for achieving shared and not reciprocally damaging goals (Arai, Pagello, and Parker, 2002).

A robot control system based on the paradigm of multiagent systems can provide rich, active entities that can adapt, represent and communicate appropriate abstractions (Vacariu, et al, 2004).

Communicating relevant information among heterogeneous units not designed to cooperate in the first place raises several challenges. The communication protocols cannot have a fix hierarchical organization as such solutions have been shown to lack flexibility for expanding and adapting knowledge and behavior for cooperating robots. The precondition is that the communication between robots is enforced by a number of component multiagents that have an efficient exchange of information based on a large spectrum of ontologies.

It has been recognized that an efficient and correct communication will improve the characteristics and the working of whatever multiagent system (Russell and Norvig, 2002).

In the design of multiagent systems, it is possible to introduce an ontological level, where specific concepts for different relevant domains of applications are described (DiLeo, Jacob, and DeLoach, 2002). This level can be an information source for multiagents.

Starting from these premises, we want to demonstrate that the use of communication at the ontological level of information representation is a reliable method to obtain a correct functioning of a system of mobile robots acting in cooperation.

Vacariu L., Chintoanu M., Lazea G. and Cret O. (2007).
COMMUNICATION AT ONTOLOGICAL LEVEL IN COOPERATIVE MOBILE ROBOTS SYSTEM.
In *Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics*, pages 455-460
DOI: 10.5220/0001620904550460

The paper is structured as follows: Section 2 shows briefly the benefits of introducing the ontological level in the multiagent systems design, to achieve the information exchange. Section 3 describes a cooperative mobile robots system for monitoring, manipulating and cleaning in a supermarket and explains the use of ontological level in communication. Section 4 reports the test conditions and results obtained in the simulation. Section 5 presents conclusions and research ideas for future work.

2 ONTOLOGY IN MULTIAGENT SYSTEM DESIGN

Ontology denotes here a common vocabulary with basic concepts and relationships of a domain (Noy and McGuinness, 2001). The domain description requires representations for types and properties of objects, and for relations between them. Software agents use ontologies and knowledge databases on ontologies as information sources.

The ontology classes describe concepts of a domain in a taxonomic hierarchy. The roles describe classes' properties and have different values for different instances. They may be restricted or not to a certain set of values. The knowledge database is created from individual instances of classes, with specific values for properties, and supplemental restrictions.

The aim of ontology integration in a multiagent system design is a description of information using the ontological level of knowledge representation. A common vocabulary of a domain allows definitions of basic concepts and relations between them to be included in the design process of a multiagent system. This creates premises to obtain a new system with new facilities, that is also more robust and adaptable.

We use the framework of MultiAgent System Engineering MASE (DeLoach, Wood, and Sparkman, 2001) that starts from an initial set of purposes and makes the analysis, design and implementation of a functional multiagent system. The ontology can be built during the analysis stage and after that is being used to further create new goals for the system. Purposes often involve parameter transmissions and consequently the ontology classes can be used as parameters.

Objects of the data model are specified as parameters in inter-agent conversations. In role refining and conversation building, that involves meta-message transmissions, that includes the type specification for transmitted parameters. The actions can use information contained in the parameter attributes because the types of parameters and attributes of types are all known. The internal variables representing purposes and conversations can be standardized with respect to the system ontology. The validity of conversations is automatically verified using parameters and variables.

Communication is achieved by inter-agent conversation. A conversation defines the coordination protocol between two agents and uses two communication diagrams between classes: one for the sender and the other one for the receiver. The communication diagram between classes is made of finite state machines, which define the states of conversation between the two participant classes.

The sender always starts a conversation by sending the first message. Any agent who receives a message compares it with all the active conversations from his list. If a match is found, the required transition is made, and the new required activities of the new state are achieved. If there is no match, the agent compares the message with all the other possible messages he might have with the sender agent, and if it finds one that matches, it will start a new conversation.

To be able to participate in conversations, the agents use information from disposition diagrams where the name, address and configurations of agents and stations are saved.

Conversations have no blocking states; all the states have valid transitions from where it is possible to reach the final state.

The transaction syntax uses UML notation:
receive_message(arg1)[condition]/action^transmit_message(arg2).

Robot systems made by autonomous mobile robots execute missions in spatial environments. The environment description using a spatial ontology will highlight entities with respect to space. The spatial ontology defines concepts used to specify space, spatial elements and spatial relations. Therefore, when we create the ontology for the multiagent system used in multirobot system, we insist on physical objects, on concepts that describe the environment using spatial localization of the components. The terms of the concept list are organized in classes and attributes, and an initial model data is elaborated. The necessary concepts of the system are specified for purpose achievement. The communication in multiagent systems uses then the concepts of the developed ontology.

3 COOPERATIVE MOBILE ROBOTS SYSTEM

To verify the effectiveness of using the communication at the ontological level in a mobile multirobot system we designed and implemented a multiagent system. The mobile multirobot system has the purpose to serve, in cooperation, a supermarket. The goals the system must provide are supervising the supermarket, manipulating objects of different types and dimensions, cleaning and giving alarm if necessary. Logging of all events and situations is also necessary.

The multirobot system is cooperative. All the robots have the goal to accomplish activities in collaboration.

3.1 Multiagent System Design

The specification for the cooperative multiagent system gives the following general objectives: to supervise the supermarket; to move in the assigned zone in the market; to identify forms or objects that do not match the knowledge about the supervised environment and notify theirs positions; to manipulate objects (pick up, move and push); to coordinate activities; to exchange information (sending and receiving); to select the appropriate agent with respect to activities (by competencies, positions in space, costs); to clean, and to log events.

The system's objectives are grouped in a hierarchy, based on the importance and connections between them (Figure 1).

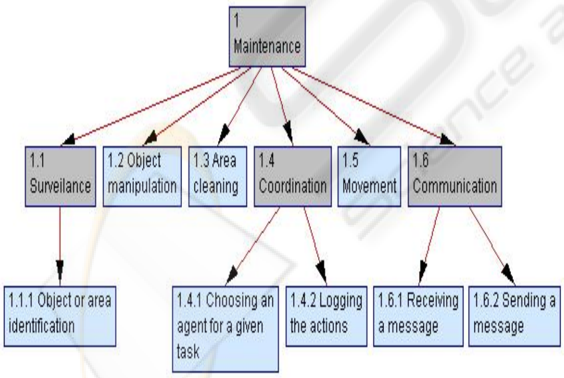

Figure 1: Objectives hierarchy.

From the initial specifications of the system, one defines the base scenarios for identifying the communication ways, like obstacle movement, zone cleaning, and intruder identification.

Using these cases, we build sequence diagrams from where the initial roles are then identified.

In the multiagent system design we highlight the defining elements of the domain described by the ontology and the inter-agent communication methodology based on using ontology concepts.

The taxonomy of the used ontology and part of the concepts are taken from the SUMO (Suggested Upper Merged Ontology) ontology (SUMO, 2006). We insist here on physical entities, with necessary add-ons for multirobot system applications. The ontology has a larger information domain than necessary for this particular application, and allows simple updates. This facility will assure their usage also in futures applications.

The ontology base concept is the *Entity*, the central node of the hierarchy. In the tree we have *PhysicalEntity* and *AbstractEntity*. We consider every concept that may be looked like an entity with spatial and temporal position as a *PhysicalEntity*. In this category, a distinction is made between *Object* and *Process*. The *Object* concept has complete presence in any moment of his life. The ontology tree is developed with concepts down to the multirobot application agent's level. *Object* can be *Group* or *Individual*. We develop the *Individual* concept with *Region*, *Substance* and *ConectedObject*.

The objectives structures and sequence diagrams were converted in roles and goals associated with them. The model of roles includes roles, everyone goals and also information about goals interactions (Figure 2).

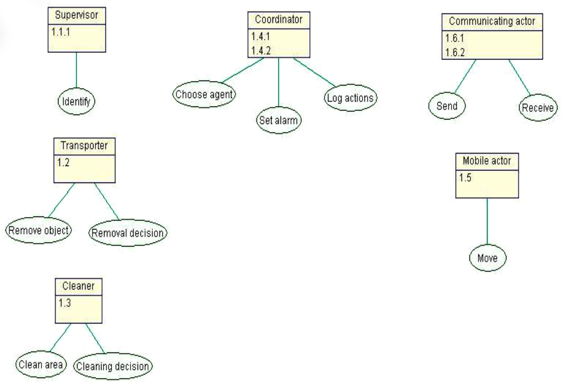

Figure 2: Roles model.

3.2 Multiagent System Conversations

From the role model, the multiagent system implements different agent types: supervisor, coordinator, transporter, cleaner, mobile agent, and communication agent. The agent's class diagram highlights the roles and the conversations between different types of agents. For every identified communication, two conversation diagrams are necessary, each with a sequence of steps. For instance, obstacle movement has conversation diagrams for supervisor, coordinator, and transporter. Every sent message has a name and content. The content can be missing if the message has only control role.

In our approach, the message content is always an ontology object. By this kind of messages we can transmit among agents any concept from the system ontology. Each time when information about a system object is exchanged, an ontology object will be in fact transmitted. The minimum information of content is the type, coordinates and dimensions of an object.

In the multiagent system built to serve a supermarket, the obstacles that must be moved can be different objects from the ontology. For instance, when the coordinator sends a message to the transporter for moving an object, this object can be taken from two conceptual categories, with the same parent *ConnectedObject*. One concept is *Artifact – Product* where the obstacle can be *Desk*, *Table*, *Chair*, *Case*, *Bin*, and the other concept is *OrganicObject*, that can be *Human* or *Plant*. The transporter will send to the supervisor a message with an object that is itself, *TransporterAgent*, and has the coordinates, self-identifier, and other information necessary for mission completion. In the ontology taxonomy, *TransporterAgent* is part of the *MobileAgent* concept, which identifies the mobile robots used in mobile multirobot system, and comes from *Artifact – Device – Robot* concepts.

The agents are instantiated and put into a network diagram. A number, type and location identify them. The built multiagent system is dynamic. Therefore in every moment new agents can be introduced. Agents are placed on the same computer or in remote computers, based on their association with mobile robots. The collective communication is provided by broadcast transmission. Every agent is a separate execution thread and has one's own port for sending and receiving messages.

By using the method of information exchange at the ontological level, it is possible to share knowledge from the entire ontology between any agents from the multiagent system. This improves the capabilities of the system and assures a reliable execution of the missions, a better adaptability and specialization to changes.

4 SIMULATION CONDITIONS AND RESULTS

The multiagent system implementation is made in Java language, with IntelliJ IDEA support (IntelliJ IDEA, 2006). Conversations structure uses agentMom (Message Oriented Middleware), component of agentTool (MASE developer) (agentTool, 2005). The application uses BroadcastServer interface, implemented by every agent or agent component. An agent can share different types of conversations. The agents and conversations are in different threads of execution. The agents run in parallel, independently on each other. All conversations are made at the same time.

The ontology classes are instantiated and transmitted in messages content. The system's agents have different knowledge from the ontology. For instance, the supervisor agent knows all kinds of objects used in conversations because it is the one who identifies the objects and decides what type they have. The coordinator has the same knowledge like the supervisor because it makes the connection with all the agents from the system. The transporter and cleaner agents know only sub-trees of ontology, those in correspondence with their activity areas.

The concepts used in conversations from ontology are *Case* for obstacle, *ConnectedObject* for dirt and *Human* for intruder.

Objects use length, width, and height dimensions. The height is used to compute the volume. The intruders don't use dimensions because only their position is important for the system. The positions of objects are implemented in the *Object* concept.

We implement a WorldInstance class with information about the types and positions of agents and objects, the minimum, maximum, and implicit dimensions, and the work area. A part of this information can be modified.

The multiagent system developed for the mobile multirobot system has been tested in simulation conditions. We built a simulation framework for sets of activities. The agents are associated with robots having different functionalities. Possible problems of synchronization and sharing resources have been solved.

Agents have windows associated with them. These are placed in tabs in main window of the application interface used for messages exchange and actions. These messages show that the ontological information is indeed exchanged. The coordinator has two supplementary windows, to manage the control of supervisors and transporters.

We chose to simulate a number of well-defined, controllable and repeatable critical ontological situations encountered in this environment. The functional situations tested were: detection of obstacles, dirt, and intruders in the supervised area, moving detected obstacle, cleaning detected dirt, moving supervisors in other areas, new transporter or cleaner agent with better capacities insertion, coordination of supervisors, coordination of transporters and cleaners, alarm activation, and actions logging.

In the environment described in Figure 3.a, the system allows detection of two cases, Square and Rectangle, by supervisors S0 and S1. S0 and S1 detect both the Square, but just one transporter is sent to move it. The transporters will move those cases that are the closest from them. The other case and dirt located outside the supervisors areas won't be moved (Figure 3.b). To detect these case and dirt, S1 supervisor is moving in another area (Figure 3.c). After moving the case by T1 and cleaning the dirt by C0, all cases are in the storehouse; transporters are in the waiting area and the cleaner remains where he finished his last action. The environment looks like Figure 3.d.

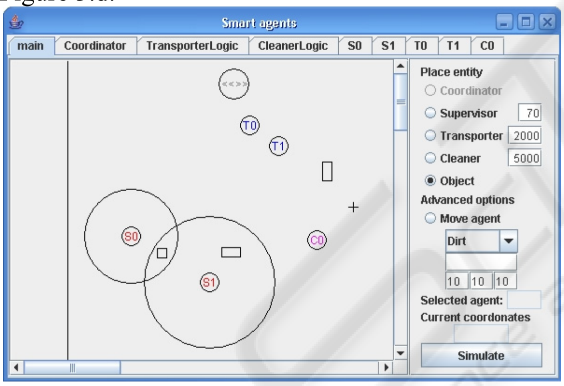

Figure 3.a: Environment.

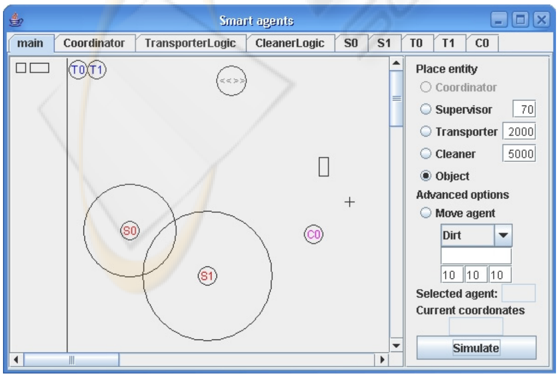

Figure 3.b: Square and Rectangle detection and moving.

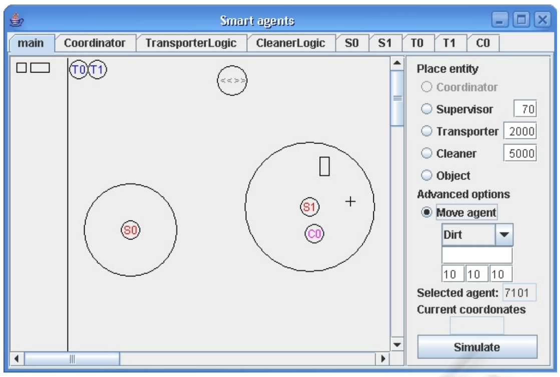

Figure 3.c: Case and dirt detection.

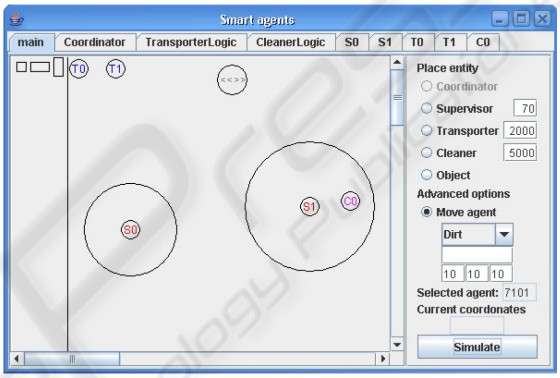

Figure 3.d: Final simulation.

Agents' windows show that the system, as designed, provided all necessary information in communications between coordinator, supervisors, transporters and cleaners. Objects from ontology were successfully exchanged and all agents correctly finished their goals, as seen in the coordinator window (Figure 4).

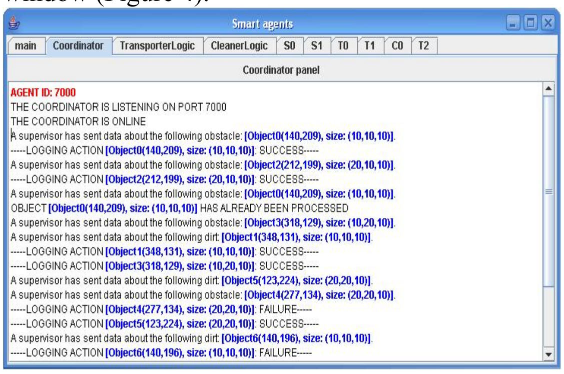

Figure 4: Coordinator window.

The action is not executed and it is logged as failure, if some information differs for some agents (especially the dimensions of objects). This implies that new agents with better capabilities must be introduced to accomplish the goals.

One other environment situation (Figure 5.a.) has the transporter T2 with larger capacity and an intruder that will be detected and after that the coordinator will start the alarm (Figure 5.b).

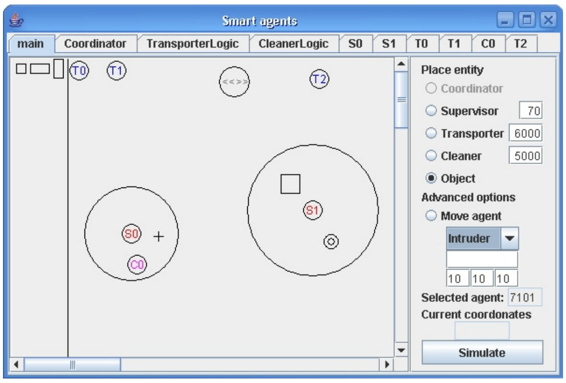

Figure 5.a: New agent and objects.

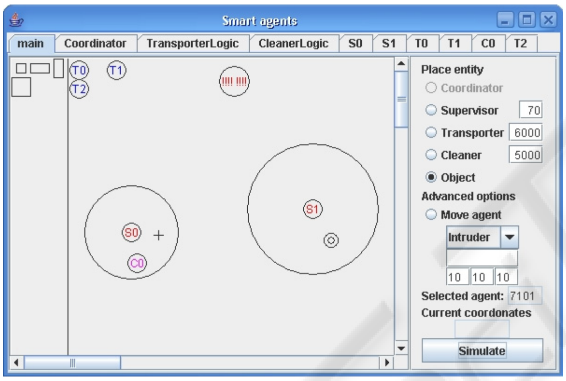

Figure 5.b: Intruder detection and alarm.

The dirt is not removed because the cleaner agent C0 is filled with the previous dirt. The new big case is successfully processed by T2.

The coordinator performs logging of all events and situations arising, for later consultation. For every action we record the time elapsed for its execution, the object that was the goal of the action together with its characteristics, the result of action, and which agent executed the action. When an action is not finished or it was instantaneously accomplished, in the log file will be record only the time of object registration.

5 CONCLUSIONS

The design of multiagent system is valid with respect to the purposes and requests of mobile multirobot system. All agents are able to accomplish successfully the missions and actions assigned to them. The system is robust and flexible.

Using the ontological information in inter-agent communications simplifies the communication process. Conversations made with ontological information are oriented to describe the spatial representation by concepts of the environment, specific to robotic systems.

The system allows efficient changes in the content of the messages among robots and proves that heterogeneous agents, having dissimilar knowledge can, by using ontology information, exchange necessary information to accomplish complex goals.

The system has been tested under simulated conditions. The positive results are leading to our next goal, to make tests under real environments, with mobile robots.

We will also test our approach in other types of missions for mobile multirobot systems.

REFERENCES

AgentTool, 2005. http://macr.cis.ksu.edu/projects/agentTool/

Arai, T., Pagello, E., and Parker, L.E., 2002. Advances in Multi-Robot Systems. In *IEEE Transactions on Robotics and Automation Magazine*. Vol. 18, No. 5, pp. 655-661.

DeLoach, S.A., Wood, M.F., and Sparkman C.H., 2001. Multiagent Systems Engineering. In: *The International Journal of Software Engineering and Knowledge Engineering*. Vol. 11, No. 3, pp. 231-258.

DiLeo, J., Jacob, T., and DeLoach, S., 2002. Integrating Ontologies into Multiagent Systems Engineering. In *AOIS-2002 Proceedings of Fourth International Bi-Conference Workshop an Agent-Oriented Information Systems*. Bologna, Italy.

IntelliJ IDEA, 2006. http://www.jetbrains.com/idea

Noy, N.F., McGuinness, D.L., 2001. Ontology Development 101: A Guide to Creating Your First Ontology. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05* and *Stanford Medical Informatics Technical Report SMI-2001-0880*.

Russell, S.J., and Norvig, P., 2002. *Artificial Intelligence: A Modern Approach*, Prentice Hall, USA, 2nd edition.

SUMO, 2006. http://protege.stanford.edu/ontologies/sumoOntology/sumo_ontology.html.

Văcariu, L., Blasko, P.C., Leţia, I.A., Fodor, G. and Creţ, O.A., 2004. A Multiagent Cooperative Mobile Robotics Approach for Search and Rescue Missions. In *IAV2004 Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*. Lisbon, Portugal.