MODEL OF KNOWLEDGE SPREADING FOR MULTI-AGENT SYSTEMS

D. Oviedo, M. C. Romero-Ternero, M. D. Hernández, A. Carrasco, F. Sivianes and J. I. Escudero
Departamento Tecnología Electrónica, Universidad de Sevilla, Sevilla, Spain

Keywords: Multi-agent system, Knowledge sharing, Knowledge spreading, Ontologies (artificial intelligence), Open systems, Software engineering, Agent communication, Automatic control system, Interoperability model.

Abstract: This paper presents a model to spread knowledge in multiagent-based control systems, where simplicity, scalability, flexibility and optimization of communications system are the main goals. This model not only implies some guidelines on how the communication among different agents in the system is carried out, but also defines the organization of the elements of the system. The proposed model is applied to a control system of a solar power plant, obtaining an architecture which optimizes agents for the problem. The agents in this system can cooperate and coordinate to achieve a global goal, encapsulate the hardware interfaces and make the control system easily adapt to different requirements through configuration. The model also includes an algorithm that adds new variables in the communication among agents and enables flow control knowledge in the system.

1 INTRODUCTION

A goal of control systems is to enable the integration of different types of devices in a scalable and flexible system. The problem is how communications among the different parts of the system is organised and optimized (Huget, 2003).

The control systems have evolved to a complex system that employs more and more equipment and sensors. The hardware diversity of the sensors and actuators greatly affects the portability of the control system and the complexity may differ from each other. A solar power plant control system (CARISMA Project) is inserted in this paper. It has a distributed input-output agent architecture to accomodate the changing requirements. This distributed intelligent agent architecture provides flexible and scalable ways to integrate the different sensors and actuators. The design goal is to create a system architecture that is general enough to support many different kinds of sensors and actuators, while being distributed and scalable.

To resolve problems in the control system it is necessary have knowledge and experience working in the field. In the case presented, the solutions to control problems will be the responsibility of different agents in the system. The agents should handle the problem domain knowledge and be able to communicate this knowledge in order to provide efficient solutions and recommendations. In addition to the system architecture, we propose a model for spreading the knowledge of agents within this architecture. This model presents an organizational scheme of agents and a global communication protocol. The model was constructed to support a wide range of control systems; however, this paper focuses on its application in the control of a solar power plant.

2 PROBLEM DOMAIN

The problem discussed here belongs to the field of distributed control systems, particularly in the problem of knowledge spreading in distributed network agents of control. An introduction to the subject and issues can be found at (Li et al., 2008), (Yang et al., 2009). Control systems based on the theory of multiagents, and restricted to specific domains have been developed. For instance, one of the early works by (Wang Junpu, 2000) discusses the feasibility of agent-based distributed hierarchical intelligent control. Another study examines the modeling of multiagent control for energy infrastructures (Sebastian Beer, 2009) that presents an agent-based control system for distributed energy resources in low voltage power grids.

Knowledge representation mechanisms primarily

Oviedo D., C. Romero-Ternero M., D. Hernández M., Carrasco A., Sivianes F. and I. Escudero J. (2010).
MODEL OF KNOWLEDGE SPREADING FOR MULTI-AGENT SYSTEMS.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Artificial Intelligence and Decision Support Systems*, pages 326-331
DOI: 10.5220/0002973803260331

used for sharing of knowledge among agents, has also been a focus of research. For instance, the use of mathematical logic, for knowledge representations and exchange among agents has been explored in (Alessio Lomuscio, 2000). However, our focus is on using ontologies for knowledge.

Within this domain, the most important aspect that we discuss concerns the spread of knowledge in control systems based on multiagents. In (Mara Adela Grando, 2006) the study of knowledge transfer and action restriction among agents in multiagent systems founded on the definition of patterns of dialogues between groups of agents, are expressed as protocols. In this paper we focus in the organization of the elements of the system and the flow of information within the system, with the aim of creating a simple and optimized model.

3 MODEL OF KNOWLEDGE SPREADING

In multi-agent systems, one important aspect is the sharing of knowledge among agents. To share knowledge it is necessary to conceptualize the problem domain and this should be common to all agents (Knowledge Representation). It is also necessary to define the processes for sharing knowledge or acquiring knowledge by an agent. The following section presents our model for the implementation of these aspects discussed.

3.1 Knowledge Representation in the Agents

Traditionally, agent knowledge representation is done through what is called Ontologies (Colomb, 2007). Ontologies define a set of elements as Predicates, Actions of agents, Concepts, Primitives, Expressions, Variables, etc. In the case of control systems, the emphasis is mainly on the concepts and actions that an agent could handle while other elements are more intended for communication tasks.

The concepts can be easily represented by data structures. In our CARISMA project, these concepts will refer to the different variables that are necessary to manage in order to make a solar panel control. These concepts can be stored locally on each agent or centralized, but in any case, the concepts must be shared by agents, so that they can communicate. The mechanisms used to update concepts can be based on simple broadcasting of messages, updating a central database accessed by the various agents to learn the concepts used or in distribute mechanisms (Sebestyenova, 2005).

The actions of the agents are modeled by expert systems (Yanping Du, 2005), using facts and rules. These rules represent real actions on devices such as reporting recommendations, faults or alarms in the system that have been located by one or more agents. Also these rules in an agent may represent the action required to share knowledge with other agents or to update their own knowledge. The mechanisms for updating expert systems agents are outlined in the next sections.

3.2 Knowledge Sharing by Agents

To spread the knowledge in a multi-agent system, it is necessary that agents are equipped with mechanisms for communication of this knowledge through the network that interconnects them. These mechanisms are usually based on communication primitives (Walton, 2003; Milner, 1994) defined in the ontology of the system.

In order to improve the reliability of the recommendations and proceedings of a multi-agent control system, we propose to label the knowledge that spreads through the network. Such labeling would be based on two concepts: the reliability of information and the reputation of the agent that has communicated the information. The *reliability* indicates the level of credibility of the received information by an agent, while the *reputation* indicates the degree of trust that the recipient agent has regarding the agent which transmitted the information. These labels can be represented as a percentage. In the transmission of information, the transmitter adds information about the *reliability* to the knowledge and labels the *reputation* for the receiver.

3.3 Knowledge Acquisition by Agents

When an agent receives a message from another agent which carries new knowledge or an update existing knowledge, there must have a process / task, designed to update the expert system of the destination agent (Dongliang et al., 2008). These processes operate directly on the agent's knowledge base, so additionally we propose an algorithm that considers the labels that are associated with knowledge: reliability and reputation.

In the algorithm the variable *newK* represents the new knowledge, *SA* is the source agent and *DA* is the destination agent, *newFb* and *fb* are the new fiability and the fiability label respectively, *rp* is the reputation label of source agent .*OldK* refers to the old knowledge stored in the destination agent. Depending on the results of that algorithm, it is determined if the knowledge update process will take place or not in the agent.If the algorithm opens the way to update the information for the agent, it also indicates the labels of reliability and reputation of the new knowledge gained.

Below is the algorithm written in pseudocode and details of its operation , as well as the set of functions that are used in it.

Algorithm 1. acquisitionKnowledge.

```
if not hasKnowledge(DA,newK)
   if acceptByReputation(DA,SA)
      newFb=newK.fb/SA.rp
      insertKnowledge(DA,newK,newFb)
   else
      discardKnowledge(DA,newK)
   endif
else
   if acceptByReputation(DA,SA)
      if(oldK.fb <= newK.fb)
         newFb=newK.fb/SA.rp
         updateKnowledge(DA,oldK,newK,newFb)
      else
         if confirmKnowledge(DA,newK)
            newFb=newK.fb/SA.rp
            updateKnowledge(DA,newK,newFb)
         else
            discardKnowledge(DA,newK)
         endif
      endif
   else
      discardKnowledge(DA,newK)
   endif
endif
```

If the information is new knowledge (*not hasKnowledge*) then the destination agent will check the reputation it has of the source agent (*acceptByReputation*, for example based on a threshold). When the new knowledge is accepted by reputation, then the algorithm gets the value of the new label of reliability (such as the division between the reliability of information and the value of reputation of the source agent) and the new knowledge is inserted in the agent's knowledge base stations (*insertKnowledge*). In other cases, new knowledge is discarded (*discardKnowledge*). If the information is an update of knowledge that is already possessed by the agent destination and is accepted by the reputation of the destination agent then there are two cases. In the first case, label reliability of old knowledge is lesser or equal to the fiability of new knowledge, in which case it updates the knowledge base of the agent (*updateKnowledge*). In the second case, the reliability of the new knowledge is not superior to what the agent already possessed, in which case the algorithm allows an alternative mechanism to be used for the acceptance of new knowledge (*confirmKnowledge*).

This algorithm provides a basic skeleton for the processes of acceptance, updating or rejection of knowledge in an agent. The number and type of labels used can be extended (but may involve increased complexity of the process).

3.4 Knowledge Spreading in the Agents Network

We have proposed this model looking to optimize the number of messages sent through the network to communicate knowledge and to establish a structure that allows some agents to perform certain recommendations or make decisions based on knowledge dispersed over different parts of the network. So, our model proposes the organization of a multiagent system in at least three layers. This is in the capabilities knowledge management and communication with agents (the fact that an agent belongs to one layer or another will depend primarily on the behaviors that it implements). figure 1 shows a diagram of the propagation model.

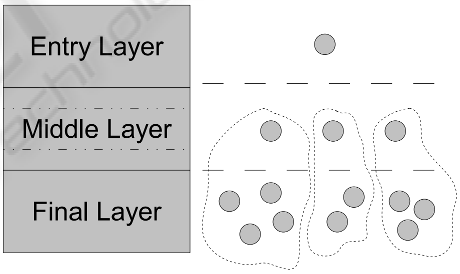

Figure 1: Model Spreading Knowledge.

As shown in figure 1, the first layer, called the *input layer* comprises all agents that allow for the entry of new knowledge in the system introduced by a user or generate changes in the knowledge base of the system from information sent by lower layer agents. The number of agents that may belong to this layer is not limited, but the simplest is formed by a single agent (decisions and recommendations centralized generator). The agents of this layer only communicate with the agents of intermediate layers, but allow direct communication with final layer agents when the response speed requirements are high (ocasionally).

The *middle layer* comprises all agents that have the ability to generate decisions or recommendations based on knowledge acquired from different points or areas of the multi-agent system and coordinate the agents belonging to lower layers. This layer can be subdivided into many middle layers as desired, allowing for scale and creating a hierarchy multiagent system for decision or recommendation generation. The number of agents in this layer is not limited. It is usually a number greater than one and less than the number of agents of the bottom layer that controls. In the communication level, the agents of this layer are able to coordinate (horizontal comunication) and disseminate knowledge in the layers in both directions (upper-layer and lower-layer).

The *final layer* comprises all terminal agents that have the ability to obtain environmental data or act on it. The number of agents in this layer is not limited, it is usually a number greater than the nummbers in the rest of layers. These agents can be organized into zones or regions of coverage controlled by a set of agents from the upper layer. The agents of this layer only communicate with the agents of the intermediate layers, but allow direct communication with agents from the input layer when the response time requirements are high (ocasionally).

figure 2 shows the different possibilities of the communication flow modeled, according to the source of knowledge to be communicated. The possibilities are: (A) Communication of new knowledge by a human user, or propagation of an action in the system from a global knowledge of system. (B) Communication of local knowledge: In this case, knowledge spreads from the final agents to agents of the *middle layer* or *input layer*. In the latter case, the spread is usually done through the middle layer, and it can be performed directly among agents from the final layer and the input layer (dotted lines) if response time requirements are high. (C) Communication of knowledge to the input layer or final layer by an agent of the middle layer, from a partial knowledge of the system.

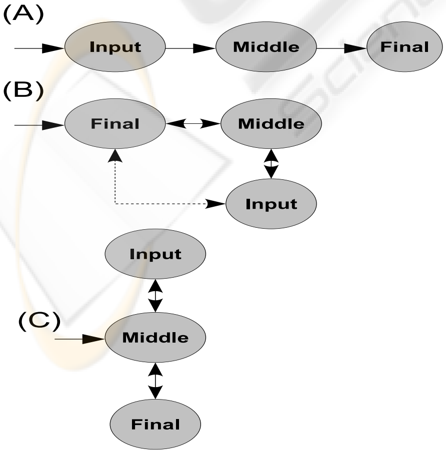

Figure 2: Spreading Knowledge Flows.

4 PROPOSED ARCHITECTURE

4.1 System Overview

Based on the described model, we propose a general system architecture composed of four types of agents: Teleoperator Agent (TA), Coordinator Agent (CA), Operator Agent (OA) and Device Agent (DA). This architecture is the basis of the CARISMA project.

The number of coordinator, operator and device agents is free (specified in the configuration of the system), while there is only one teleoperator agent. Communication among agents is restricted: TA can communicate with any agent of the system (and vice versa), while CAs and OAs will have specific information about what other agents they can communicate with. The DAs may only communicate with the OA to which they are assigned. This configuration allows us to define flexible areas, by supporting different communication channels among agents living in the system, which can lead to the possibility of overlapping in these areas.

An example of the network topology including the three zones is shown in figure 3.

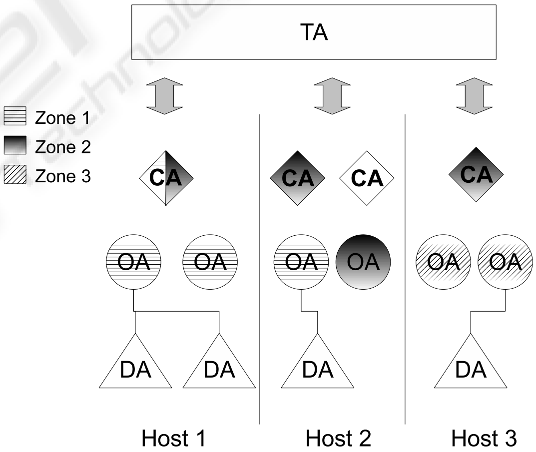

Figure 3: General architecture for CARISMA.

The network can expand or shrink according to the number of the solar power plants and the complexity of the control system. The architecture of individual agents is based on the paradigm Belief-Desire-Intention (Huiliang and Ying, 2005). The Teleoperator Agent is the entry point into the system, providing a user interface that allows configuration, deployment and knowledge input to the platform. The Coordinator Agents goals are to coordinate global solutions for a state of failure or alarm detected from multiple points in different areas. The Operator Agents are responsible for controlling the various DAs, and if they detect failures or local alarms in accordance with information received from the DAs, they communicate them to the rest of the system. The Dispositive Agents are hybrid agents (Cognitive and Reactive), that have the ability to alert the OA or act directly in case of changes in the state of a device. Each DA will have a concrete implementation intended to obtain data from a particular sensor or perform actions on a given actuator. An example of communication among OAs and DAs is shown in figure 4.

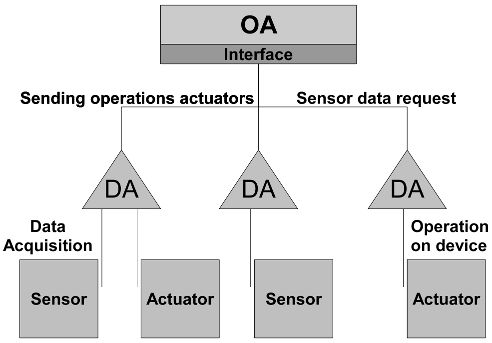

Figure 4: Communication architecture for OAs and DAs.

In terms of hardware, there are no restrictions on the number or type agents that can reside in a device or style of devices that compose the system, but these have to be capable of computation. Generally, the agent node device consists of an embedded system with support for various transmission technologies (RF, Ethernet, Bluetooth, ...). One agent node is attached various sensors and actuator devices. The sensors include thermal sensors, humidity sensors, CO2 sensors, sensors for signal from solar plant appliances and various intelligent meters such as solar irradiation and video control (Sivianes et al., 2008). The attached actuators include various valves, motors, and switchers for heating system, ventilation, humidity control, screen control, etc.

4.2 Application of Model

When applying the model proposed in section 3 to the architecture discussed in section 4, an automatic control and decision support system which is very simple, reliable and scalable can be implemented. For instance, figure 5 shows an example of spreading knowledge in CARISMA, for the B case seen in figure 2 (generation of knowledge in the final layer). OA1, OA2, OA3 and DA are in the final layer of the system. CA represents the middle layer and TA of the input layer.

OA3 does not create any communication, and it is represented in the figure as an example of agents belonging to the same area but through which no knowledge flows.

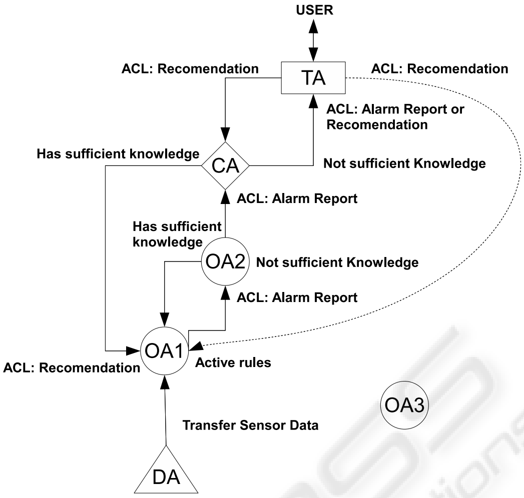

Figure 5: Example of spreading knowledge in CARISMA.

The flow of information starts in the DA1 agent, which communicates the new information acquired by a sensor to OA1. This fact produces rule activation, which implies the activation of an alarm to be communicated to the OA2 in the area. If OA2 has the knowledge to respond to such an alarm, it then advises the OA1, otherwise, this alarm must be communicated to the middle layer, represented by the coordinator agent in its area. Then, if the CA can respond to the alarm generated in the system, it will communicate directly to OA1 (in case of action) or to TA (in case of recomendation). Otherwise it will report the alarm to the next upper layer which, in this example, is the input layer, represented by a teleoperator agent. Finally, if the TA has sufficient knowledge, it can respond directly to OA1 (if imminent action is needed) or propagate the new information through the system, by the reverse route. Otherwise, the TA requests information from the user to respond to the alarm.

Note that information travels in phases throughout the system and if the necessary knowledge is not available, then a human is requested. This allows for the optimization of the information or knowledge exchanged in the system, and easily increases the knowledge base of system, with scalability.

5 CONCLUSIONS

Traditionally, control systems have had difficulties as far as their design because they must meet high requirements. Using our model in the design of a multiagent-based control system, we can obtain many benefits, such as simplicity. Based on the organization of agents in layers exposed in our knowledge spreading model, we can easily program control systems.

Other problems in control systems are scalability and flexibility: our model does not limit the number and types of agents and inclusion / exclusion of layers is possible. It also allows for dynamic configurability, so we can dynamically change parts of the system since agents can move from one layer to another, or new types of agents can be added. It is even possible to dinamically add new layers with certain restrictions, only by changing the behavior of agents. With regard to the autonomy and intelligence, the agents of the final layer can perform the input and output actions without the intervention of the central controlling computer.

Finally, our model also introduces an optimization of communication of knowledge among agents in the system. The layering allows for the design of a hierarchical system, which leads to a minimization of the exchanged messages among agents. Additionally, in the process of acquiring knowledge by an agent, the model introduced an algorithm that permits the control of information flows in the system, does so by using the concepts of reliability and reputation. This algorithm adds a mechanism to the system that allows agents to provide solutions or recommendations in a transparent and intelligent mode.

ACKNOWLEDGEMENTS

The work described in this paper has been funded by the Consejería de Innovación, Ciencia y Empresas (Junta de Andalucía) with reference number P08-TIC-03862 (CARISMA Project).

REFERENCES

Alessio Lomuscio, M. R. (2000). Intelligent agents vi. agent theories architectures, and languages. In *A Spectrum of Modes of Knowledge Sharing between Agents*. Springer, Berlin.

Colomb, R. (2007). *Ontology and the Semantic Web*. IOS Press, Nieuwe Hemweg 6B, 1013 BG Amsterdam, The Netherlands, 1nd edition.

Dongliang, L., Kanyu, Z., and Xiaojing, L. (2008). Eca rule-based io agent framework for greenhouse control system. In *ISCID '08. International Symposium on Computational Intelligence and Design*. IEEE Conference Proceeding.

Huget, M.-P. (2003). *Communication in Multiagent Systems: Agent Communication Languages and Conversation Policies*. Springer, New York, 1nd edition.

Huiliang, Z. and Ying, H. S. (2005). A parallel bdi agent architecture. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*. IEEE Conference Proceeding.

Li, H., Karray, F., Basir, O., and Song, I. (2008). A framework for coordinated control of multiagent systems and its applications. In *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*. IEEE Conference Proceeding.

Mara Adela Grando, C. D. W. (2006). Cooperative information agents x. In *Specifying Protocols for Knowledge Transfer and Action Restriction in Multiagent Systems*. Springer / Heidelberg, Berlin.

Milner, R. (1994). *Communication and Concurrency*. Prentice Hall International (UK) Ltd., New York, 2nd edition.

Sebastian Beer, M. T. (2009). Information technologies in environmental engineering. In *MACE Multiagent Control for Energy Infrastructures*. Springer, Berlin.

Sebestyenova, J. (2005). Decision support system for modelling of systems and control systems design. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce,*. IEEE Conference Proceeding.

Sivianes, F., Romero, M., Hernandez, M., Carrasco, A., and Escudero, J. (2008). Automatic surveillance in power system telecontrol applying embedded and multi-agent system technologies. In *ISIE 2008. IEEE International Symposium on Industrial Electronics*. IEEE Conference Proceeding.

Walton, C. D. (2003). Multi-agent dialogue protocols. In *Proceedings of the Eighth International Symposium on Artificial Intelligence and Mathematics, Fort Lauderdale*.

Wang Junpu, Chen Hao, X. Y. L. S. (2000). An architecture of agent-based intelligent control systems. In *Proceedings of the 3rd World Congress on Intelligent Control and Automation*. IEEE Conference Proceeding.

Yang, J., Yang, L., Zhao, T., and Jia, Z. (2009). Automatic control system of water conservancy project model based on multi agent. In *WKDD 2009. Second International Workshop on Knowledge Discovery and Data Mining*. IEEE Conference Proceeding.

Yanping Du, J. Y. (2005). Intelligent agent-based expert system architecture for generating work plan in marshalling station. In *IEEE International Conference on Granular Computing*. IEEE Conference Proceeding.