

Multiple intelligences in a MultiAgent System applied to telecontrol

D. Oviedo ^{*}, M.C. Romero-Ternero, M.D. Hernández, F. Sivianes, A. Carrasco, J.I. Escudero

Departamento de Tecnología Electrónica, Universidad de Sevilla, Seville, Spain

Keywords:

Agents
MultiAgent System
Multiple intelligences
Photovoltaic energy
Remote control
Expert Systems
Neural Network
Bayesian Network
Control systems
Solar panels

A B S T R A C T

This paper presents a control system, based on artificial intelligence technologies, that implements multiple intelligences. This system aims to support and improve automatic telecontrol of solar power plants, by either automatically triggering actuators or dynamically giving recommendations to human operators. For this purpose, the development of a MultiAgent System is combined with a variety of inference systems, such as Expert Systems, Neural Networks, and Bayesian Networks. This diversity of intelligent technologies is shown to result in an efficient way to mimic the reasoning process in human operators.

1. Introduction

The main feature for the correct functionality of a control system is the expert knowledge and how it is applied by humans. These concepts should be applied heavily in the development of these systems but they are often forgotten or minimized in order to obtain reliability. To improve these systems, it is necessary to establish mechanisms of intelligent human-like self-control.

MultiAgent Systems enable suitable models to be built for the prediction of complex and dynamic systems in real time (Saleem, Nordstrom, & Lind, 2011). Inference engines attempt to mimic the behavior of the human brain by using various types of algorithms and strategies (Dopico, De La Calle, & Sierra, 2008). The combination of both these technologies should therefore allow the construction of systems capable of solving problems satisfactorily without previous knowledge of all the variants of solutions.

The work discussed here belongs to the field of distributed control systems, particularly in the automatic telecontrol of solar power plants (Li, Karray, & Basir, 2010; Yang, Yang, Zhao, & Jia, 2009). Furthermore, control systems based on MultiAgent theory, and restricted to specific domains have been developed. For instance, one of the early works (Junpu, Hao, Yang, & Shuhui, 2000) discusses the reliability 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.

Regarding the different possibilities to build an inference engine, studies and applications have achieved excellent results for control systems based on Expert Systems (Liebowitz, 1997). These solutions provide a fast output and precise, and decrease state transitions in the real environment to control. Other options in inference systems might include Neural Networks (Cochocki & Unbehauen, 1993) or Bayesian Networks (Costaguta, Garcia, & Amandi, 2011), to reinforce the initial heuristic rules of any control system based on this kind of logic. Finally, another significant option is the use of inference systems based on Fuzzy Logic (Leondes, 1999).

MultiAgent control system architectures have been considered to reduce the computational complexity and manage the huge amount of distributed data and coupling problems among many subsystems. In many cases, implementations include only one intelligent aspect (Gadallah & Hefny, 2010; Ouidad, 2006), and often in a superficial way. This is due to the fact that intelligence and learning are very complex features to implement at software level. The inherent complexity of distributed systems causes a lack of interest in their implementation, due to requirements and reliability in such responses.

Within this domain, the most important aspect that we discuss concerns the application of real and useful intelligence models in conjunction for control systems based on MultiAgents. Studies Lorenzi, Bazzan, Abel, and Ricci (2011) and Miyashita and Rajesh (2010) use assumptions in the recommendation or coordination techniques in the making-decision, being discarded inference systems for their requirements. In this paper we focus in the orga-

* Corresponding author. Tel.: +34 954554324.

E-mail addresses: oviedo@dte.us.es, mcromerot@us.es (D. Oviedo).

nization of the elements of inference and the flow of information within the system, with the aim of creating an optimized and intelligent control system that meets all requirements exposed.

The purpose of this research is to determine the best combination of the points indicated, and to provide a viable solution through optimizing and ensuring the use of distributed artificial intelligence technologies. The result of this research is the CARISMA system, which is a new MultiAgent System application for the development of integrated systems and automatic control with distributed intelligence. This software is designed for application in the field of industrial control systems for facilities based on solar photovoltaic energy sources.

This paper is organized as follows: Section 2 shows an overview of our system, while in Section 3 the proposed architecture for a Multi-Agent System applied to a control system of a solar power plant is set out. In Section 4, the implementation and integration of the inference systems are presented together with the MultiAgent System developed. Section 5 shows the tests performed and the results obtained. Finally, the concluding remarks are found in Section 6.

2. System overview

The combination of certain technologies, to which reference is made in the introduction, leads to the integration of inference engines in the MultiAgent System. This combination is intended to provide the agents with the individual and collective intelligence necessary to resolve problems common to the entire system.

The system has been developed within the so-called "CARISMA Project: MultiAgent System for Remote Control of Solar Photovoltaic Power Plants". The objective is to control and monitor solar panel farms: in an automated way whenever possible and, the cases when it is not possible, to provide human telecontrol operators with control recommendations. To this end, small hardware devices are distributed with associated sensors and actuators in various areas of a solar farm in order to do this. This MultiAgent System employs the set of devices in order either to make decisions of automated control or to send recommendations to technicians of the solar plant based on the data and knowledge available.

2.1. Platform, environment and language development

We have made a study of the existing platforms, environments and development languages in order to facilitate the development and implementation of a MultiAgent System and the integration of several inference systems. Currently there are several platforms, frameworks and libraries that can help towards the development of MultiAgent Systems (Weyns, Parunak, Michel, Holvoet, & Ferber, 2005). These tools minimize development time and enable work to be carried out under accepted standards in MultiAgent System development.

Each of these platforms offers certain features according to the requirements of diverse applications. From among all the platforms available for MultiAgent System development, JADE (2004) has been selected since it encompasses both the features (standardization and use of generic language) required for our purpose.

3. MultiAgent System architecture

Based on the philosophy of the development of the JADE platform an architecture for MultiAgent System of the CARISMA project on the JADE platform has been created. This architecture is in accordance with the required specifications for a control system (Oviedo et al., 2010, chap. 1).

In order to design this architecture, the methodology proposed by the JADE developers (Bellifemine, Poggi, & Rimassa, 2001) has

been followed in a flexible way in order to minimize the costs involved in the initial system architecture specification.

3.1. Types of agents defined in the system

From the study carried out for the modeling of intelligence, learning and socializing, the following agents have been defined in our system:

- *Teleoperator Agent*

This is an agent who holds overall control of the platform. The main functions are to configure various aspects of the system, such as: coverage areas, which agents belong to each zone, interface for access to other agents in the system, and interface for human user access. In terms of making decisions and offering recommendations, this is the agent that has global knowledge of the platform.

- *Coordinator Agents*

The responsibility of these agents is to coordinate global solutions for an alarm situation or fault in a particular area. Examples of tasks assigned to these agents include communication failures and recommendations of solutions to other agents in this area.

- *Operator Agents*

These agents are responsible for controlling several Sensor-Device Agents that are assigned by the human operator, or for taking an action on a Sensor-Device Agent. They provide a number of communication mechanisms to communicate faults to other agents (coordinators or operators).

- *Sensor-Device Agents*

Reactive agents responsible for obtaining data from sensors and performing actions on the actuators. These agents are unique and customized to the type of device to be monitored and controlled.

In addition to these four agents, in the system a fifth agent functionality and platform boot, called "Remote Agent" (RA), exists. This agent is responsible for creating and operating the structure of agents in different physical systems that make up the software platform (embedded systems, servers, etc). The agent's life cycle is limited to the system boot, creating a temporary container when the functions are performed, and it disappears along with its container when the functions have been completed.

3.2. MultiAgent System overview

The designed architecture is intended to integrate the various types of agents described: Teleoperator Agent (TA), Coordinator Agent (CA), Operator Agent (OA), and Sensor-Device Agent (SDA). The number of Coordinator Agents, Operators and Sensor-Devices remains unlimited. It is possible to define the number of agents of each type in the system startup and remove or create them dynamically with the system running. Moreover, the number of Teleoperator Agents is limited to a single agent whose existence in the system is compulsory.

An example of the network topology defined is shown in Fig. 1. In this example, three zones or areas of communication have been defined, although the network may expand or shrink according to the number of solar panels to control or the complexity of the system control.

The Teleoperator Agent constitutes the starting point to the system, and provides a user interface that allows global configuration of the platform (shown in Fig. 2). Additionally, this interface enables the overall management of knowledge in the system. Coordinator Agents perform the semi-global coordination of solutions for fault or alarm conditions detected from multiple points in different areas of the system. Operator Agents are responsible for monitoring the

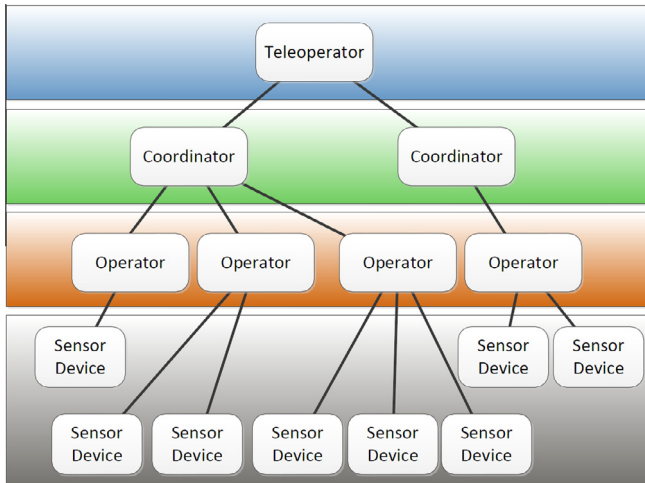


Fig. 1. Example of architecture of agents in CARISMA.

Sensor-Device Agents on common tasks and for detecting faults and local alarms according to information received by the SDA. In addition, Operator Agents can report detected faults and alarms to the remaining system agents through Coordinator Agents.

Individually, each agent presents a general system architecture based on the paradigm of Desires, Beliefs and Intentions (Rao, 1995). Multiple intelligences have been implemented for agents by using different inference systems. Sensor-Device Agents are hybrid. They have reactive abilities to enable them to operate in tasks, alerts for direct detection of failures, and actions over controlled devices (sensor readings and operations on the actuators). Each SDA has a specific implementation to extract data from a particular sensor or to perform actions on a specific actuator.

In terms of hardware, there are no restrictions on the number and type of agents that can be located in a device. Furthermore,

there are no restrictions in the type or functionality of these devices in the system. In any case, the devices that host agents should have sufficient computational capacity to run agents on the JADE platform. These electronic devices are often embedded systems with capabilities for various transmission technologies (Radio Frequency, Ethernet, Bluetooth, etc), and can be assigned to control one or several sensors or actuators. The sensors may be from different types, such as temperature, humidity, CO₂, and radiation. Other types of specific sensors involve control sensor signals concerning parameters which should be analysed in a solar power plant (voltage and intensity in the solar photovoltaic panel, battery monitoring, etc). The actuators may include control valves, on/off cooling systems, and electrical circuit drivers.

4. Intelligent control based on inference models

Once the structure of our MultiAgent System has been defined, as well as its basic elements and possible communications that can occur among agents, then it can be analysed to provide agents with capabilities of intelligence and learning.

There are three objective inference systems that have been integrated into agents to provide them with the multiple intelligences:

- An inference system based on an Expert System to provide the agents with the capacity of logical inference.
- An inference system based on Neural Networks to provide the agents with the capacity of statistical inference.
- An inference system based on Bayesian Networks to provide the agents with the capacity of probabilistic inference.

Logical inference allows a reliable monitoring of actions and decisions made by agents. Statistical inference allows behavior patterns to be found that improve and optimize certain system tasks, while probabilistic inference allows the provision of recommendations for certain problems with uncertain data.

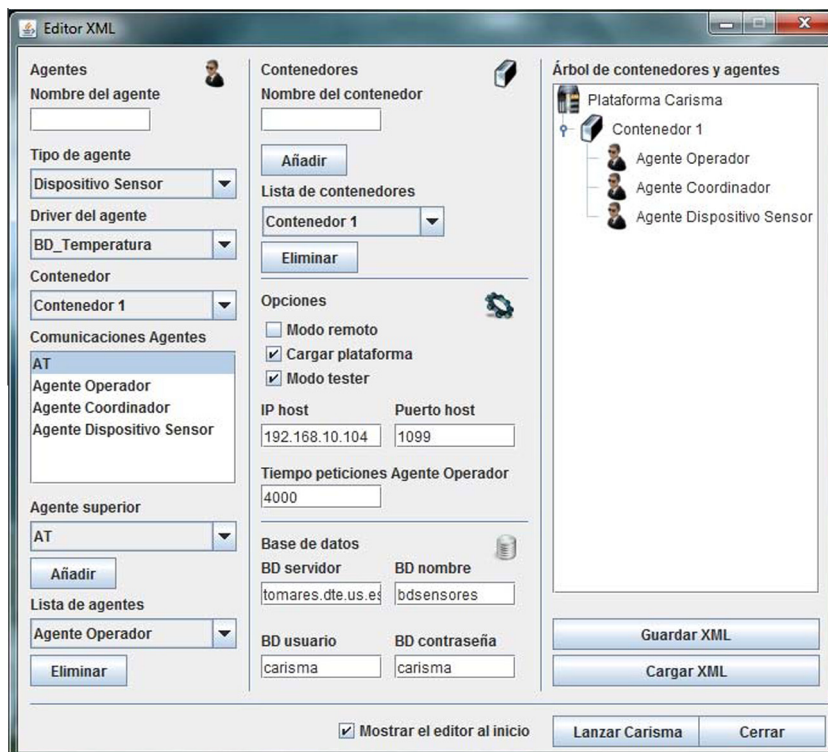


Fig. 2. Global configuration interface of CARISMA.

4.1. Intelligent control with Expert Systems

As a result of the increasing difficulty in problems to be solved by a MultiAgent System, then the availability of systems that independently enhance decision making and solutions is required. The introduction of a MultiAgent inference system helps to solve this problem, but it results in software of a more complex nature. In order to avoid this problem initially, the easiest and fastest inference system to integrate into a MultiAgent System is an Expert System.

In our implementation, an Expert System has been integrated in each agent, which manages their own knowledge base (rules and facts) and allows them to make logical decisions autonomously. In order to implement the Expert System in each agent, a rule-based system named *Drools* (2013) is used. This production system uses an advanced implementation of the “Rete algorithm” (Forgy, 1982), thereby facilitating and improving the response times.

Additionally, we have defined a set of rules and facts for the control of a solar plant in *Drools* (by extension, this is omitted in this article). To this end, a technique based on plain text files with extension “.drl” and “.dsl” has been used that defines the language domain, thereby making it possible to employ natural language with rules. This technique facilitates maintenance by end users of the application (telecontrol operators).

Moreover, it is worth bearing in mind that one of the features of *Drools* is that rules are read every time they have to be executed. This enables them to be changed even with the system running, which allows the dynamic modification of the agent’s knowledge base. This feature combined with the application of a learning algorithm based on labeling (Oviedo et al., 2010) and applied about knowledge managed for experts in power plants provides an evolutionary learning in the Expert System. It allows the gradual adaptation of the system to the needs raised by the environment.

The developed user interface (shown in Fig. 3) allows the rules associated with the agents to be changed manually. Once the user has modified all the desired rules using the rule editor, the Teleoperator Agent sends a message whose predicate is “deleteGlobalKnowledgeExchange” for each rule removed, and a predicate “AddGlobalKnowledgeType” for each new rule. Both predicates include the rule to be changed and the type of agent that has to make the change. When an agent receives the message, the agent behavior can result in a modification or insertion of the rule in its own knowledge base. If the agent type corresponds to the type specified by the message, or if it can return the message to spread it to the agents from the lower layers, then the message flows through the hierarchy of agents.

In addition to the user interface discussed, there are different ways as to how *CARISMA* sends data to *Drools* to be validated. For example, as shown in Fig. 4, it is possible that when an SDA reads data, then it can be sent directly to the rule engine, and in case of the existence of a rule trigger, an alarm will be generated to send the data to higher levels in the hierarchy. Whenever a member of the hierarchy receives data, it is sent to *Drools* to perform the corresponding actions. Such actions may differ, since the rules differ for each type of agent.

4.2. Intelligent control with Neural Networks

Initially, the Neural-Network-based model was defined for its implementation and integration into the MultiAgent System developed in order to provide an optimized solar tracking system. The objective of this Neural Network is to provide predictions of upcoming coordinates that a certain solar panel must use for performance improvement. It depends on environment variables, previous records and solar position (Oviedo et al., 2013).

Once patterns that will improve performance by the Neural Network have been found, the system requires the conclusions to be communicated to the Expert System implemented in *CARISMA*. The Expert System is responsible for carrying out actions that lead to optimal states. For the implementation of the integration with the Expert System, it was necessary to validate the Neural Network model through simulations and develop the final implementation of the network. We have chosen the development framework for Neural Networks called *Neuroph* (Sevarac, Goloskokovic, Tait, Carter-Greaves, & Morgan, 2013). In *Neuroph*, the major classes correspond to the basic concepts of a Neural Network, such as artificial neuron, layer neurons, neuron connections, weight, transfer function, input function, and learning rule. *Neuroph* supports common Neural Network architectures, such as multilayer perceptron, with backpropagation, Kohonen networks, and Hopfield networks.

It was decided to use a fully connected multilayer perceptron Neural Network with the training algorithm “Backpropagation with Momentum” (Hertz, Krogh, & Palmer, 1991), which allows temporal predictions similar to recurrent Neural Networks.

The Neural Network has received thirteen input parameters: Azimuth and Zenith solar panel coordinates, solar time UTC, Azimuth and Zenith sun coordinates, day and year, solar radiation (Kw/m^2), temperature, humidity, wind speed, atmospheric pressure, weather conditions (sunny, cloudy, rain, etc) and, finally, energy performance (kWh). The output parameters correspond to the Zenith and Azimuth coordinates recommended by the Neural Network for the solar panel.

Input data must be normalized in order to be used for the network. Normalization of the data was performed using theoretical minimum and maximum according to where our study (Seville, Spain) was completed. For those cases where one of the input parameters lies outside the specified target range, the system does not make use of the Neural Network to calculate the new coordinates, but instead the Expert System is responsible for making the decision directly. Finally, the function transfer selected is the sigmoid function because the range of the input data is normalized between 0 and 1, and this transfer function not only allows the output to be restricted, but it is also differentiable.

The simulation process enabled a set of different network parameters (maximum error, learning rate, momentum, impulse, etc) to be defined, the network to be trained by a preset maximum pattern training theory, and its behavior under different tests to be verified. The number of iterations required to train the network and the minimum error were evaluated in these results. The result of the simulations was the choice of the network, as shown in Fig. 5. This network of eight neurons in the hidden layer obtained the best results on the tests performed, and shown minimal errors in most cases, and a rapid rate of learning.

4.2.1. Integrating a Neural Network and an Expert System for final decision-making

The Expert System provides the necessary information to the Neural Network and, additionally, serves as a control element in the learning of the Neural Network using feedback. In all cases, the responsibility for the final actions carried out in the system lies with the Expert System. The Neural Network’s function is that of adaptive control, and it seeks the best adaptation for the system to achieve its best performance.

In this way, the Neural Network calculates the optimal movement patterns for the solar panels to maximize energy production, by analysing the system’s responses to small disturbances generated by the Expert System. Note that each type of agent has its own Neural Network controlling the set of variables in terms of its level of knowledge.

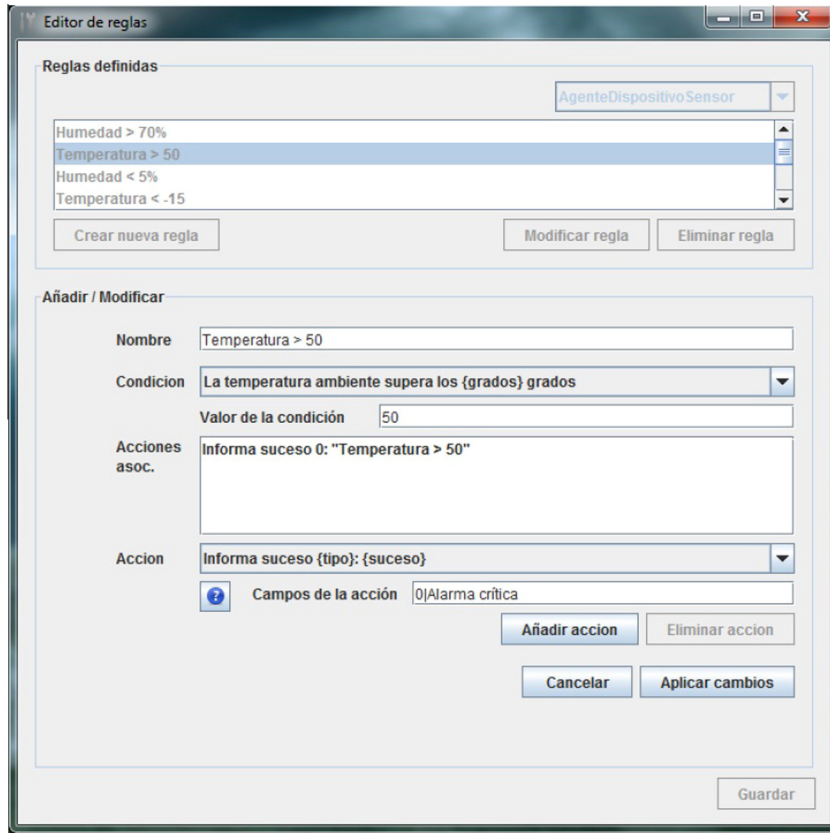


Fig. 3. Interface management rules for DROOLS Expert System integrated into an agent in CARISMA.

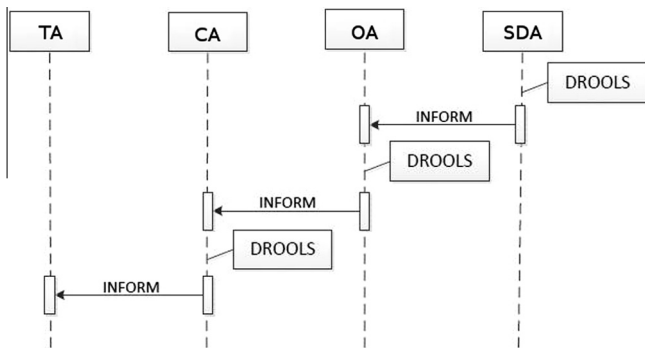


Fig. 4. Example of execution of DROOLS during the spread of knowledge.

4.3. Intelligent control with Bayesian Networks

In order to add a type of probabilistic reasoning to our MultiAgent System, we have developed a Bayesian model to determine the lifespan of a battery controlled by the MultiAgent System CARISMA. Once the Bayesian Network is developed, it is integrated with the Expert System. As with Neural Networks, the ultimate responsibility for final decisions lies with the Expert System, based on the probabilistic values supplied by the Bayesian Network.

In this way, as shown in Fig. 7, after obtaining the probability of the unknown variables, this information is supplied to the Expert System, which has a set of rules to determine an action based on the probability values of these variables.

At any time, the probabilistic information provided by the Bayesian Network of each agent can be analysed by other systems. It should also be borne in mind that each type of agent has its own Bayesian Network, which controls the set of variables in terms of its level of knowledge.

Fig. 6 shows how the Neural Network interacts with the Expert System in CARISMA to perform suitable training for the movement on solar panels under certain environmental circumstances.

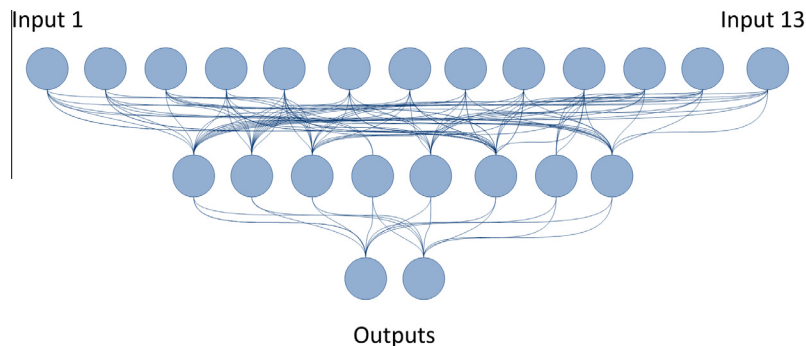


Fig. 5. Final topology selected for the Neural Network in CARISMA.

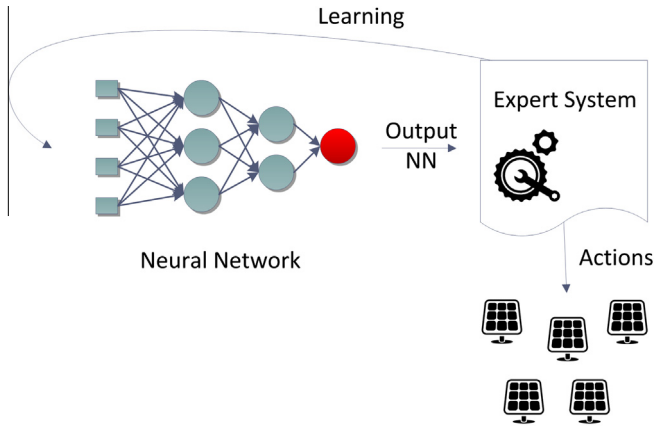


Fig. 6. Integration model for the Expert System and Neural Networks in CARISMA.

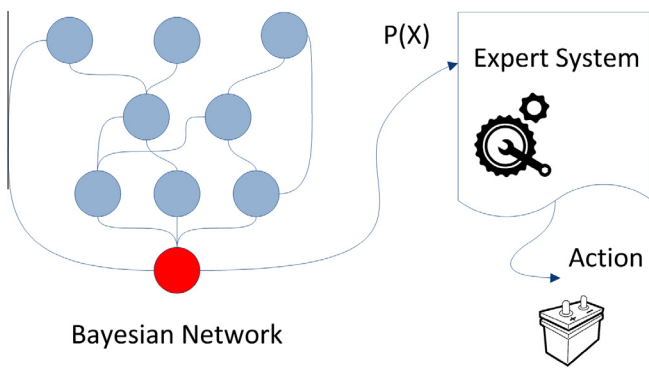


Fig. 7. Integration of the Bayesian Network with the Expert System in CARISMA.

The modeling for the prediction of the lifespan of a battery is based on a learning model of event patterns and information provided by sensors about the physical state of the battery. Three tasks were carried out for the implementation of the network: (1) definition of probability tables; (2) implementation of the Bayesian Network using specific software; and (3) integration of the Bayesian Network by defining new control rules for the Expert System.

The definition of probability tables is performed by taking into account previous studies on battery lifespan (Oviedo et al., 2014) (by extension, omitted in this article). From that study observable variables have been defined: temperature, intensity, and voltage of the battery. Unobservable variables correspond to elements or processes that can occur in a battery but cannot be read by a sensor (density, load, depth of discharge, etc). The output variable corresponds to the probability of the analysed battery lifespan.

In order to carry out the task of implementing the Bayesian Network, an external library was employed to achieve stability and robustness through an established working system: SMILE (Loboda & Voortman, 2013). In Fig. 8, the main window of the application with the developed Bayesian Network is shown.

The battery used consists of 6 stationary SUNLIGHT 20PzS100-150A vessels (150 Ah C-100) (TechnoSun, 2012). This system of accumulation is for the storage of energy produced by the photovoltaic farm during sunshine hours, for later consumption. These Pb-acid stationary batteries are the most common in photovoltaic systems. In addition to the batteries, a controller is included for TriStar MPPT Morningstar solar panels (Morningstar, 2012), which is able to control the battery charging by detecting the maximum power point, MPPT, for independent photovoltaic panels.

4.3.1. Integrating a Bayesian Network and an Expert System for final decision-making

Finally, the integration with the Expert System consists in the definition of a new set of rules for the interpretation of the moment when there is a high probability of the battery lifespan coming to an end. In this case, the agent sends an alarm to the Teleoperator Agent with a warning about the imminent shortage.

On the other hand, the input of the Bayesian Network is connected to the output of the Expert System in order to enable the system to generate a response for a specific event input. These events have to be able to change the Bayesian Network inputs so that its probability tree can be restored. However, due to the Bayesian Network implementation in SMILE, this cannot be carried out directly without using the Expert System to update the network input. To this end, new rules have been laid down for the Expert System which involve setting the Bayesian Network input values to new values defined by the Expert System itself.

On this way, in a scenario where the Bayesian Network makes decisions, it makes no sense for the actions to be directly linked to outputs. Instead, the new system sets the corresponding value in the Bayesian Network, preventing the probability trees from being updated.

5. Measurements and results

The studies performed are focused on the overall performance of the developed MultiAgent System CARISMA, as well as on the elements that provide intelligence to agents: Expert Systems, Neural Networks and Bayesian Networks. This research has studied these last elements in depth, since they constitute the main object of this contribution.

Various configurations and scenarios of CARISMA have been established for testing. The scenarios have been chosen in order to carry out both a study of performance of the overall system, and as well as an analysis of certain parameters for each integrated inference system, which are of interest in the research accomplished here.

5.1. Technique for MultiAgent System analysis

In the study of the CARISMA system, use has been made of an external measuring system for MultiAgent Systems on the JADE platform called PeMMAS (Carrasco et al., 2014). Additionally, this system has been combined with the use of internal implementations of specific and precise measures, focused on the study of the implemented inference systems.

The PeMMAS system allows the performance of a MultiAgent System developed within the JADE platform to be studied and measured. This tool is another MultiAgent System that is deployed along with the MultiAgent System under study. With this tool, data can be analysed concerning the use of system resources, mean times of global messages in communications, as well as processing times of the behaviors. It can also be used for other feasible measures external to the system agents. The data collected by the measuring agents of PeMMAS are processed to generate various reports that allow the analysis of the MultiAgent System under study.

5.2. General scenario of system operation

A general structure of the system is shown in Fig. 9, as it covers all areas of solar energy distributed across several solar farms.

A set of various configurations and scenarios has been defined from this general architecture in order to comprehensively test the performance of CARISMA. All scenarios were created and tested in the laboratory, where hardware devices are necessary to

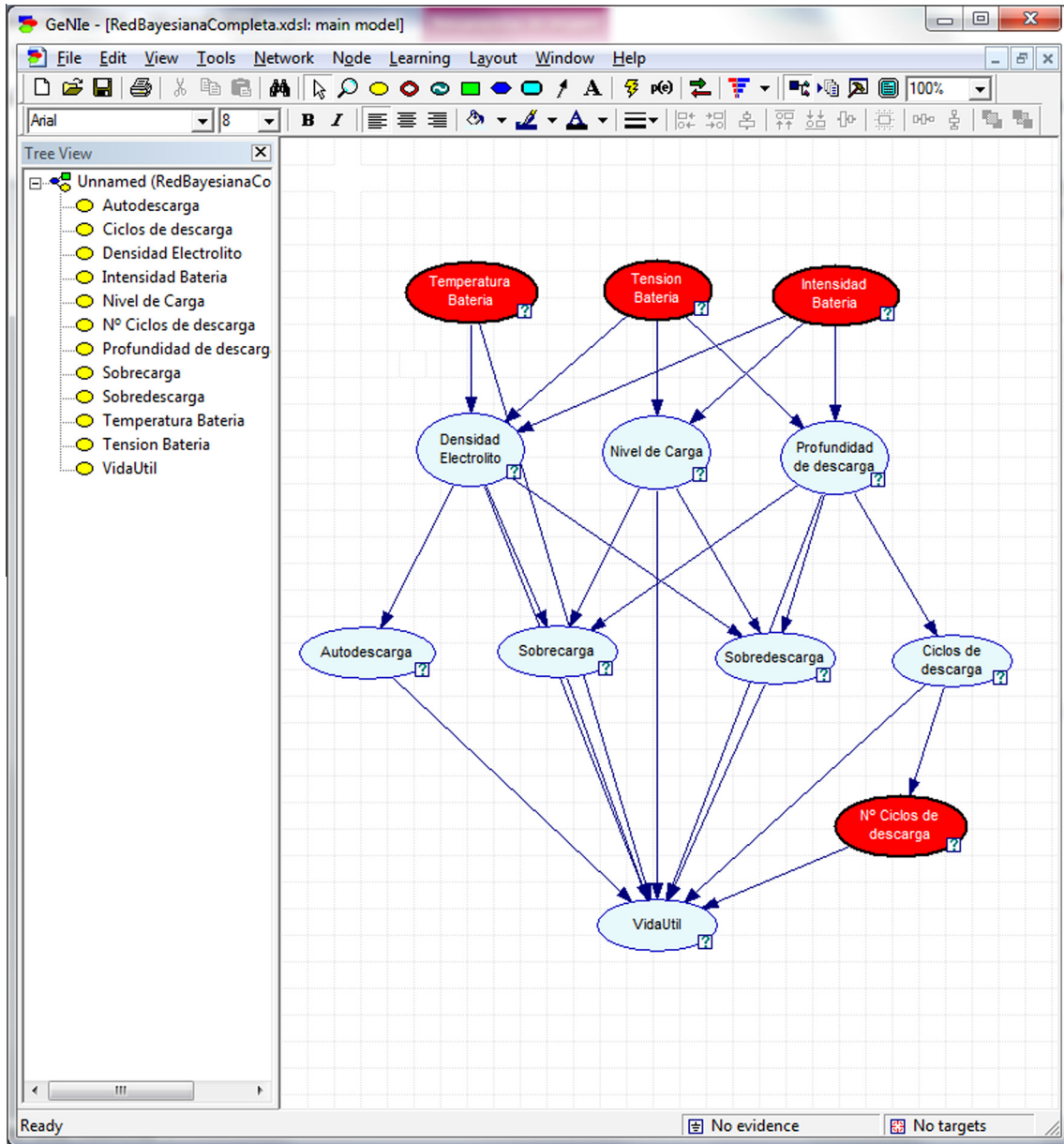


Fig. 8. Bayesian Network for calculating the battery lifespan in SMILE.

simulate the operation of a photovoltaic system correctly. In total, four possible scenarios with a variety of objectives have been studied, although numerous variants have been developed and studied in order to determine limits on the system and verify its operation.

Tests and measurements have enabled interpretation of the developed architecture, as well as the obtainment of conclusions and proposals for improvements in the MultiAgent System and in its integrated inference systems.

5.2.1. Hardware used in the tests

As a test environment, a free space was available that could be dedicated to placing the infrastructure of solar panels, and a laboratory where control devices, such as computers and embedded systems, of the solar panels and the global energy system could be located.

The set of devices along with their technical information includes:

- Sensor network based on motes “SquidBee” (SquidBee, 2013) and connected by ZigBee (2014) technology. This network obtains environment information for the agents. These motes are the first “open mote” in the world and have been developed by Libelium (2014), and have the following characteristics:
 - Arduino + XBee module
 - Open source mote
 - 9 V battery
 - Easy software programming
 - 12 I/O digital lines
 - 6 analog input lines
 - 5 PWM analog output lines
 - USB connection to PC (windows, linux and mac compatible)
 - Wireless communication, based in XBee module (ZigBee)
 - Sensors:
 - * Temperature
 - * Humidity
 - * Brightness

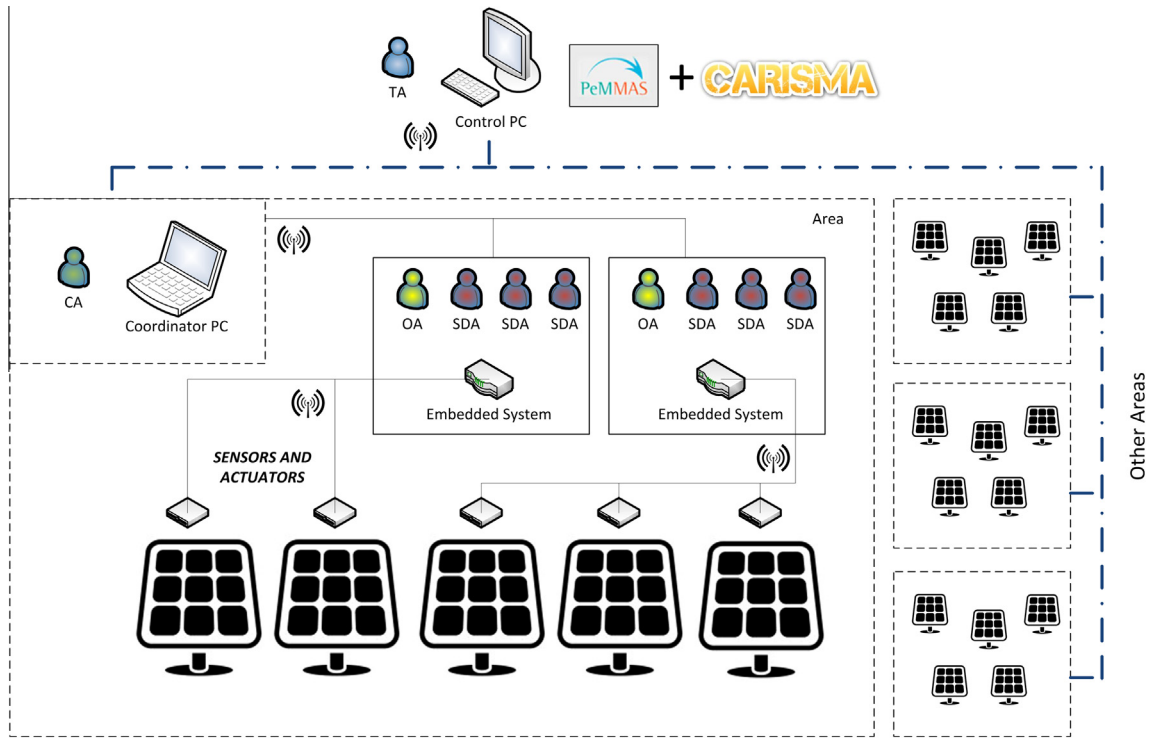


Fig. 9. General scenario of system operation in CARISMA.

- Embedded systems based on AMD Geode chip [AMD Geode chip \(2013\)](#), designed to accommodate field control agents. Technical specifications:
 - CPU 233 MHz, AMD Geode SC1100 CPU (fast 486 core)
 - 128 MB SDRAM
 - Linux operating system based on Debian
 - Connection to network via Ethernet or Wi-Fi
- Controller for solar panels Tristar MPPT. This enables system monitoring for solar panels, log data and for set parameters. It has wireless network through Ethernet.
- Sunlight Batteries 20PzS 100, 2 V – 100 Ah.
- Photovoltaic solar panels: Model Kaneca G-EA060, with nominal power of 60 Wp.
- Sunlight inverter 1100 W.

The various devices can be connected via Ethernet, and WiFi, or Zigbee in the case of motes designed to collect data from sensors.

5.3. Tests and results

In this section the four specific scenarios defined for the study of parameters related to performance and resolutions by inference systems integrated in CARISMA are detailed. The inference systems are implemented into various types of agents according to their specifications and requirements. Although it is possible to integrate inference systems developed in all agents, their use is restricted to only those agents that require it:

- Every agent implements a workspace within the Expert System. In each machine or hardware device that contains an agent, an instance of the inference engine “Drools” runs, and each agent has its own database knowledge and workspace.
- Coordinator Agents integrate the developed Bayesian Network when they are responsible for controlling batteries in a solar farm.

- Operator Agents integrate the developed Neural Network when they are responsible for controlling the movement of solar panels.

5.3.1. Tests on Expert Systems

In order to study the operation of CARISMA, once behaviors are implemented and integrated into the Expert Systems in agents, a specific scenario arises where the system performance and response times are analysed. In this scenario, a solar farm consists of five areas where all areas have the Teleoperator Agent in common. Additionally, each area has its own Coordinator Agents and Operator Agents. This scenario is shown in [Fig. 10](#).

Various basic performance tests have been executed in order to verify the correct implementation of Expert Systems into agents. These tests, which were carried out on different scenarios, resolved minor issues of implementation and enabled a stable system to be attained without any problem concerning alarm resolutions and recommendations offered by the Expert System.

The percentage of successful recommendations for the developed system is very high; almost 95% compared with expectations. This high percentage is due to the application of an Expert System, since once it is well modeled in its rules and facts, it becomes very reliable owing to the predefined business rules. Erroneous recommendations (average of 2%) or uncertain recommendations (between 2% and 6%) result from the knowledge propagation model implemented in the system which can lead to inconvenient rejections and inconvenient changes in rules. This propagation model compensates for any possible inconsistencies. It allows a progressive increase of knowledge into the agents, thus avoiding unnecessary traffic through the data network.

In terms of time measurements, in these tests an average response time of one second by inference engine Drools has been observed, as can be seen in [Fig. 11](#). Response time depends on several factors, such as the amount of information handled by the agent in question. In tests, Sensor-Device Agents employ fewer rules and facts because they are essentially reactive and focus on

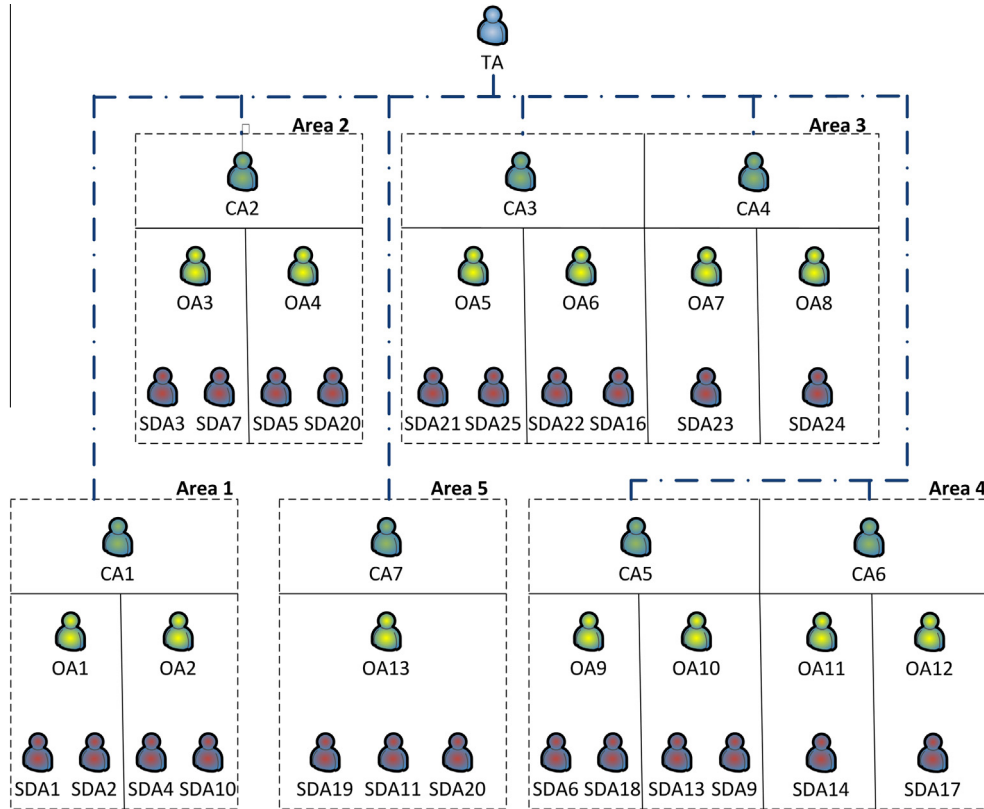


Fig. 10. Testing scenario for Expert Systems.

direct implementation tasks and reading data. The quantity of knowledge handled by an agent is reflected in the response times. The amount of knowledge handled by an agent varies depending on the role of the agent in the system architecture (OA, CA, and finally TA).

In Fig. 11, the average response time for agents handling a high quantity of knowledge is shown. These times are a bit slow for real-time applications although they are suitable for control tasks where response time requirements are fast but not immediate (below 2 s in TA inferences and below 200 ms for SDA). In any case, the quantity of knowledge rises incrementally in the hierarchy of agents, thereby adjusting the system response time according to the needs of a particular control system.

Finally, as shown in Fig. 12, the times required by Expert Systems to learn are very low. This is due to the inference engine optimization offered by Drools. The inclusion of new facts or rules is immediate but there is a small time increase in changes and rejections of rules as a result of the requirement for previous analysis.

5.3.2. Tests on Neural Networks integrated with the Expert System

Research of the built-in-agent Neural Network has been performed on a solar farm composed of five areas. Each Operator Agent is responsible for the movement of a solar panel or a group governed by the same engines. The scenario for testing is shown in Fig. 13. Our implementation meets the following requirements:

- Operator Agents integrate a Neural Network in order to determine new moves to be made by the solar panels under its control. Use of the Neural Network is limited only those cases where the performance has decreased considerably from that expected (<10% of the expected performance). Otherwise, the Expert System governs the motion of the solar panels, according to the theoretical tracking.

- The information needed to supply the Neural Network is obtained from Sensor-Device Agents (SDA) in the area, which communicate such data to Operator Agents (OA) for inclusion in the Neural Network.
- Responsibility for final decisions for actions based on such monitoring processes lies with Operator Agents (automated actions).
- When the system detects a decrease of energy performance obtained in an area, the Teleoperator Agent (TA) may seek to optimize performance in said area. Control and communications are governed by Coordinator Agents (CA) in an area.

The main purpose of this Neural Network is to provide adaptive control for the movement of solar panels in order to maximize the energy obtained. Therefore, the set of tests is performed to verify the proper operation of the Neural Network, as well as the results obtained in the overall system. Such tests have focused on the analysis of energy efficiency obtained during the same day. For this purpose, we have arranged a history of performance measures obtained for various days of the year in the position where the panels are located. Such measures have three variants: with fixed solar panels, solar panels with classic solar tracking (based on theoretical solar position), and maximum performances for various hours of the day.

Once the different tests have been carried out, we obtain an average measure of the most relevant values in the system for the set of all agents (Oviedo et al., 2013). In these tests, the system has obtained an average increase of 3% in performance compared to that obtained with a traditional solar tracking system. The coordinates of following movement predicted by the Neural Network are those expected in more than 10% of cases and close to those expected in more than 25%, and also obtain a high number of movements (approximately 15%) that are not in target range for performance optimization. Regarding the consumption by the

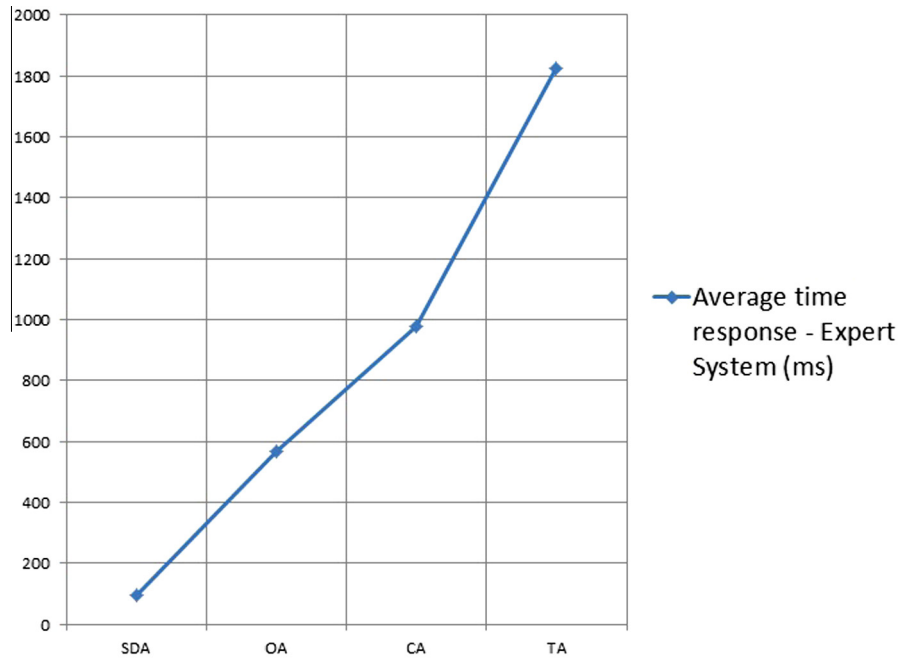


Fig. 11. Average response time of the Expert System in an agent of CARISMA.

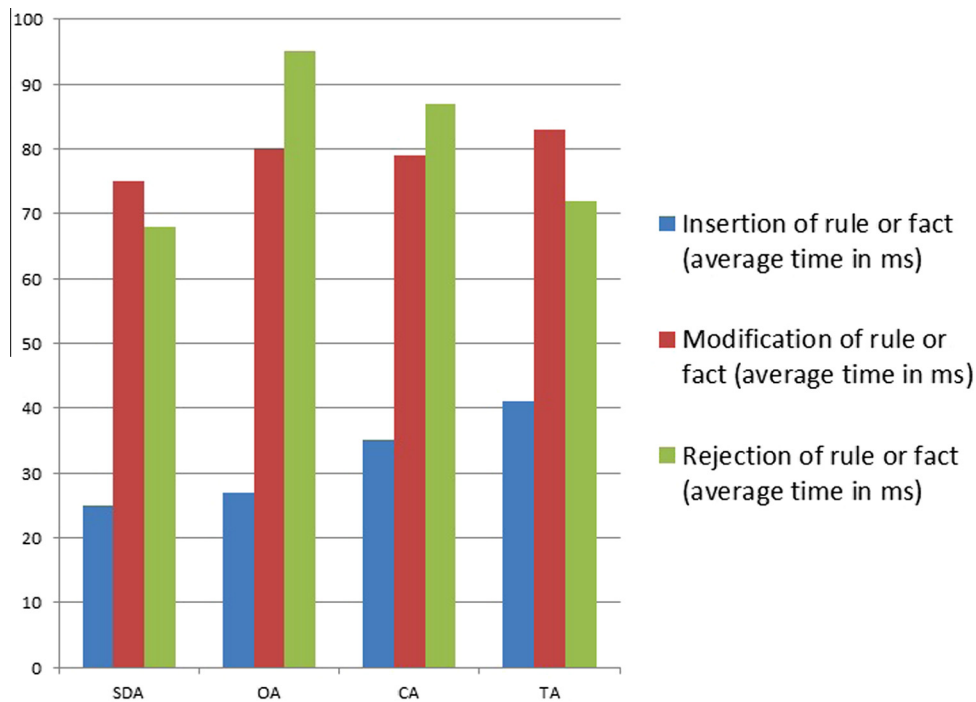


Fig. 12. Average time required by Expert System on learning tasks.

CPU, the Neural Network has a low effect on the system, with a consumption less than 0.5%, because the use is limited and controlled by the Expert System. However, in the case of RAM, consumption is somewhat higher, reaching an average consumption of 3 megabytes.

5.3.3. Tests on Bayesian Networks integrated with the Expert System

A specific scenario is assumed where it is possible to verify proper control of the solar battery lifespan in order to study the behavior of the Bayesian Network implemented and integrated within the system agents. In this scenario, the solar farm consists of three areas, with a common Teleoperator Agent and a specific

number of its own Operator and Coordinator Agents. Furthermore, each area has its own battery monitored by the system.

The test scenario proposed is shown in Fig. 14 and meets the following requirements:

- Coordinator Agents implement a Bayesian Network to perform monitoring.
- The information needed to supply the network is obtained by the Sensor-Device Agent in the area, which communicates such data to its Coordinator Agent for inclusion in the Neural Network (the communication can be made through the Operator Agents).

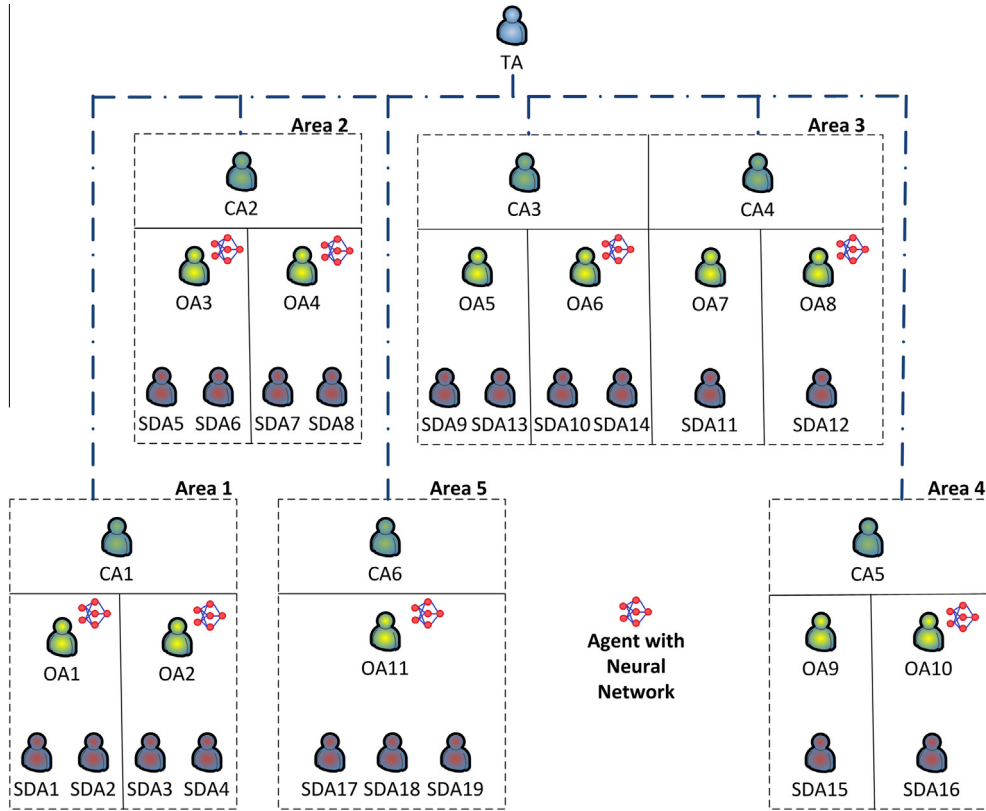


Fig. 13. Testing scenario for Neural Networks.

- Responsibility for final decisions for action based on such monitoring lies with the Teleoperator Agent, Coordinator Agents should supply recommendations for the subsequent actions.
- If more than one Coordinator Agent coexist in an area, each agent implements an independent Bayesian Network and sends its recommendation to the Teleoperator Agent, which should select only one of them in the case of contradiction.

The required time to check the response of the system to obtain an assessment of the lifespan of a battery controlled by the system would be no less than two years in real conditions. Therefore we have developed a stand-alone application that changes the value of the parameters involved, as well as their rate of change. This application simulates the behavior of the system covering two years within a few hours of simulation. From simulations, we have

obtained the average results shown in Fig. 15. These values are maximum probability values obtained by the Bayesian Network for a given set of circumstances over time. If the value of the lifespan probability is 0, it is considered that battery is exhausted, while a value closer to 1 means improved battery status.

We compared the results obtained with the expected results and observed that the hit rate was high. It can be concluded that in the extreme tolerance of the battery ($> 2K$ cycles, $T < -10^{\circ}C$, $T > 50^{\circ}C$), the system shows a high probability that battery lifespan is coming to end, whereas if we approach the optimal storage conditions ($10^{\circ}C < T < 40^{\circ}C$ and $V < 11,8 V$), the probability indicates a longer lifespan.

Additionally, a set of tests is performed to check the impact of using Bayesian Networks on the consumption of disks and processors in the system. The precision used in the tests is of the order of

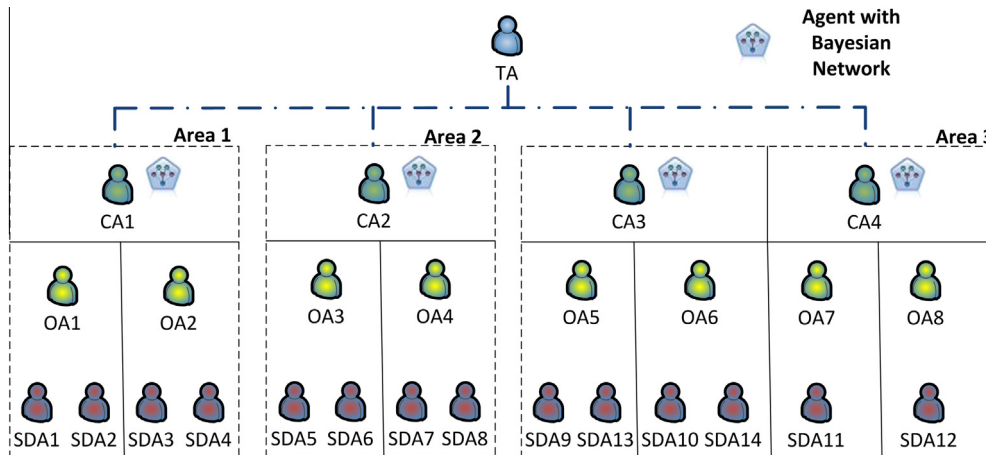


Fig. 14. Testing scenario for Bayesian Networks.

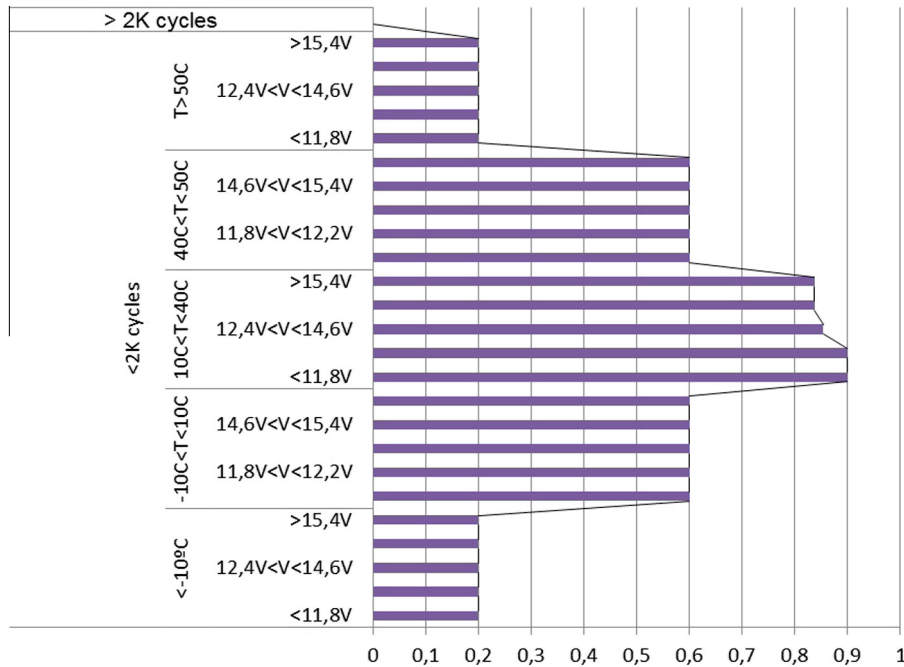


Fig. 15. Probability of battery lifespan obtained by the Bayesian Network developed.

nanoseconds. Regarding the consumption of CPU, Bayesian Networks do not have any impact. In fact, the consumption range lies between 0.5% and 1.5% of the CPU when Bayesian Network is presented. Regarding the RAM, no case exceeds the megabyte consumed. Therefore the impact of the network is negligible.

6. Conclusions and future works

A complete MultiAgent System has been developed for monitoring and managing a solar power plant. The system agents have been equipped with multiple artificial intelligences through the combination of Expert Systems, Neural Networks, and Bayesian Networks. Once the system is completed, it is verified through tests and simulations where the responses obtained by the system are adequate, with only minor human intervention.

The use of different intelligences by combining the three types of inference offers various practical advantages in a MultiAgent control system. From a point of view of development, the use of multiple intelligences enables a system that is able to assimilate knowledge progressively, as opposed to traditional control systems, where it is necessary to encode all the knowledge in advance. It enables a new level of abstraction and thus minimizing development time.

Another important practical application is to enable solving several types of problems, under one centralized system of knowledge. In our research, the system is able to solve logical, statistical and probabilistic problems in the same problem domain and system architecture. So, it is not necessary other modules or hardware requirements and the interfaces are commons.

Therefore, a distributed intelligence within a MultiAgent System enables a control system with enough capacity to meet its objectives without human intervention. The CARISMA system is a self-control system for solar power plants, which in practice does not requires human support once the learning has been carried out.

Regarding the conclusions from the point of view of theory obtained from research is remarkable that our combination of inference models ensures system reliability and time responses. Our combination model can create reliable control systems and

with resource requirements it is able to select the inference systems used and their goals properly. In general, it is important to conclude that providing agents with intelligent capabilities is possible and even feasible where time and resource requirements are imposed by control systems.

In addition, this type of implementations closes the practical definition of an agent to its theoretical definition. Agents are intelligent and autonomous in practice, and not just distributed software modules.

Future lines of work include apply the combination of different inference mechanisms for modeling and simulation of human behavior from the perspective of the field of Psychology. The first motivation is to exploit the power of the MultiAgent systems as a research tool for developing emotional intelligence and social intelligence. And secondly and more specifically, with the idea of supporting in health-care and studying emergent behavior in society related to the use of new technologies.

Acknowledgments

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

- Advanced Micro Devices, Inc. (2013). AMD Geode Processor Family website. <<http://www.amd.com/us/products/embedded/processors/geode-lx/pages/geode-lx-processor-family.aspx>>.
- Bellifemine, F., Poggi, A., & Rimassa, G. (2001). Developing multi-agent systems with JADE. In C. Castelfranchi & Y. Lespérance (Eds.), *Intelligent agents VII agent theories architectures and languages. Lecture notes in computer science* (Vol. 1986, pp. 42–47). Berlin, Heidelberg: Springer.
- Carrasco, A., Hernandez, M., Romero-Tertero, M., Sivianes, F., Oviedo, D., & Escudero, J. (2014). PeMMAS: A tool for studying the performance of multiagent systems developed in JADE. *IEEE Transactions on Human-Machine Systems*, 44(2), 180–189.
- Cochocki, A., & Unbehauen, R. (1993). *Neural networks for optimization and signal processing*. John Wiley & Sons, Inc.
- Costaguta, R., Garcia, P., & Amandi, A. (2011). Using agents for training students collaborative skills. *IEEE Latin America Transactions*, 9(7), 1118–1124.
- Dopico, J. R. R., De La Calle, J. D., & Sierra, A. P. (2008). *Encyclopedia of artificial intelligence*. Information Science Reference.

- Drools. (2013). Drools – the business logic integration platform. <<http://www.jboss.org/drools/>>.
- Forgy, C. L. (1982). Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19(1), 17–37.
- Gadallah, A., & Hefny, H. (2010). A novel multiagent system based on dynamic fuzzy cognitive map approach. In *10th International conference on intelligent systems design and applications (ISDA)* (pp. 254–259).
- Hertz, J., Krogh, A., & Palmer, R. G. (1991). *Introduction to the theory of neural computation*. Santa Fe Institute studies in the sciences of complexity: Lecture notes. Addison-Wesley Pub. Co.
- Jade. (2004). *JADE: JAVA agent development framework*. Tech. rep., Telecom Italia Laboratory, <<http://jade.tilab.com/>>.
- Junpu, W., Hao, C., Yang, X., & Shuhui, L. (2000). An architecture of agent-based intelligent control systems. In *Proceedings of the third world congress on intelligent control and automation* (Vol. 1, pp. 404–407).
- Leondes, C. (1999). *Fuzzy theory systems: Techniques and applications*. Fuzzy theory systems (Vol. 1). Academic Press.
- Libelium Comunicaciones Distribuidas. (2014). Libelium official website. <<http://www.libelium.com>>.
- Liebowitz, J. (1997). *The handbook of applied expert systems*. Taylor & Francis.
- Li, H., Karray, F., & Basir, O. (2010). A framework for coordinated control of multi-agent systems. In *Innovations in multi-agent systems and applications – 1. Studies in computational intelligence* (Vol. 310, pp. 43–67). Berlin, Heidelberg: Springer.
- Loboda, T.D., & Voortman, M. (2013). SMILE and GeNle website. <<http://genie.sis.pitt.edu/>>.
- Lorenzi, F., Bazzan, A. L., Abel, M., & Ricci, F. (2011). Improving recommendations through an assumption-based multiagent approach: An application in the tourism domain. *Expert Systems with Applications*, 38(12), 14703–14714.
- Miyashita, K., & Rajesh, G. (2010). Multiagent coordination for controlling complex and unstable manufacturing processes. *Expert Systems with Applications*, 37(3), 1836–1845.
- Morningstar. (2012). Datasheet – Morningstar Tristar MPPT. <<http://www.morningstarcorp.com/en/tristar%20mppt>>.
- Ouidad, K. (2006). Interacting with reasoning in a MAS integrating a RBS. In *ICTTA '06 information and communication technologies* (Vol. 2, pp. 3001–3006).
- Oviedo, D., Romero-Tertero, M. C., Hernández, M., Carrasco, A., Sivianes, F., & Escudero, J. (2010). Model of knowledge spreading for multi-agent systems. In *ICCS conference proceedings, international conference on enterprise information systems, Funchal, Madeira, Portugal* (pp. 326–331).
- Oviedo, D., Romero-Tertero, M., Carrasco, A., Sivianes, F., Hernandez, M., & Escudero, J. (2013). Multiagent system powered by neural network for positioning control of solar panels. In *39th Annual conference of the IEEE industrial electronics society, IECON 2013* (pp. 3615–3620).
- Oviedo, D., Romero-Tertero, M., Carrasco, A., Sivianes, F., Hernandez, M., & Escudero, J. (2013). Simulation and implementation of a neural network in a multiagent system. In *The eighth international conference on intelligent systems and knowledge engineering, ISKE 2013*.
- Oviedo, D., Romero-Tertero, M. C., Carrasco, A., Hernandez, M. D., Sivianes, F., & Anaya, P. (2014). Inferring battery lifespan in solar farm multi-agent telecontrol system. *IEEE Intelligent Systems*, submitted for publication.
- Oviedo, D., Romero-Tertero, M. C., Hernández, M., Carrasco, A., Sivianes, F., & Escudero, J. (2010). *Architecture for multiagent-based control systems* (Vol. 79). Berlin, Heidelberg, Alemania: Springer (pp. 97–104, chap. 1).
- Rao, A. S. (1995). Georgeff, BDI agents: From theory to practice. In *Proceedings of the first international conference on multi-agent systems (ICMAS-95) San Francisco* (pp. 312–319).
- Saleem, A., Nordstrom, L., & Lind, M. (2011). Knowledge based support for real time application of multiagent control and automation in electric power systems. In *16th International conference on intelligent system application to power systems (ISAP)* (pp. 1–8).
- Sebastian Beer, M. T. (2009). Information technologies in environmental engineering. In *MACE – multiagent control for energy infrastructures*. Berlin: Springer.
- Sevarac, Z., Goloskokovic, I., Tait, J., Carter-Greaves, L., & Morgan, A. (2013). Neuroph, <<http://neuroph.sourceforge.net/>>.
- Libelium Comunicaciones Distribuidas. (2013). SquidBee datasheet. <<http://libelium.com/squidbee/upload/c/c1/SquidBeeDataSheet.pdf>>.
- TechnoSun. (2012). Datasheet – 20PZS100-150A. <<http://www.technosun.com/es/descargas/SUNLIGHT-OZPS-ficha-EN.pdf>>.
- Weyns, D., Parunak, H. V. D., Michel, F., Holvoet, T., & Ferber, J. (2005). Environments for multiagent systems state-of-the-art and research challenges. In *Environments for multi-agent systems* (pp. 1–47). Springer.
- Yang, J., Yang, L., Zhao, T., & Jia, Z. (2009). Automatic control system of water conservancy project model based on multi agent. In *Second international workshop on knowledge discovery and data mining, WKDD* (pp. 349–352).
- ZigBee alliance. (2014). ZigBee official website. <<http://www.zigbee.org>>.