Available online at www.sciencedirect.com

ScienceDirect

Computers in Industry 58 (2007) 313–328

COMPUTERS IN INDUSTRY

www.elsevier.com/locate/compind

Agents-based design for fault management systems in industrial processes

Mariela Cerrada [a,*], Juan Cardillo [b], Jose Aguilar [c], Raúl Faneite [d]

[a] Universidad de Los Andes-Escuela de Ingeniería de Sistemas, Departamento de Sistemas de Control, CEMISID, Núcleo La Hechicera, Facultad de Ingeniería, Piso 2, Ala Sur., Mérida 5101, Venezuela

[b] Universidad de Los Andes-Escuela de Ingeniería de Sistemas, Departamento de Sistemas de Control, Núcleo La Hechicera, Facultad de Ingeniería, Piso 2, Ala Sur., Mérida 5101, Venezuela

[c] CEMISID, Universidad de Los Andes-Escuela de Ingeniería de Sistemas, Departamento de Computación, Núcleo La Hechicera, Facultad de Ingeniería, Piso 2, Ala Sur., Mérida 5101, Venezuela

[d] PDVSA-CRP, Punto Fijo 4102, Venezuela

Received 28 April 2005; accepted 13 July 2006

Available online 2 November 2006

Abstract

Industrial necessities claim global management procedures integrating information systems in order to manage and to use the whole information about controlled processes and thus, to assure a good process behaviour. Fault management and maintenance are vital aspects in industrial management, in this sense, maintenance systems should support decision-making tools, new maintenance approaches and techniques, the enterprise thinking and flexibility. In this work, a reference model for fault management in industrial processes is proposed. This model is based on a generic framework using multi-agent systems for distributed control systems; in this sense, the fault management problem is viewed like a feedback control process and the actions are related to the decision-making in the scheduling of the preventive maintenance task and the running of preventive and corrective specific maintenance tasks. A particular methodology permitting the conception and analysis of the agent systems is used for the agents design. As a result, a set of models describing the general characteristics of the agents, specific tasks, communications and coordination is obtained.

Keywords: Maintenance; Fault management; Multi-agent system; Automation

1. Introduction

Automation is an important aspect allowing for the improvement of the industrial processes performance. In this sense, automation systems aim to dynamic supervision of operating variables and, from this information, the calculation of control signals. In this sense, industrial necessities claim global management procedures in order to integrate the information systems of both enterprise and operation levels into the control systems and use this information for optimization purposes.

A five level's architecture is considered, in general, for industrial automation schemes [1]: process, local control, supervision, planning and administrative management, each one with specific tasks. In general, the couple model–control has constrained suppositions about the process' dynamic and its causes are not thoroughgoingly modelled, thus, in case of not well working, the process performance should be deteriorated. These modelling supposition put up with the development of fault management systems proposing fault detection, diagnosis and prediction tasks, and preventive maintenance plans; these tasks living together in the supervision and control levels [2–5]. In [6], a multi-agent based automation model is proposed, taking into account the general objects of the production processes: production planning, production factors management, processes control, fault management and abnormal situations management. The models describing the processes control and abnormal situations management have been developed in [7,6], respectively. The agents-based reference model for distributed control systems (DCS) in [7], has

* Corresponding author. Tel.: +58 274 2402983; fax: +58 274 2402979.
E-mail addresses: cerradam@ula.ve (M. Cerrada), ijuan@ula.ve (J. Cardillo), aguilar@ula.ve (J. Aguilar), faneiterj@pdvsa.com (R. Faneite).

0166-3615/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.compind.2006.07.008

provided a generic conceptual framework permitting to see any system like a feedback control systems. In fact, the abnormal situations management model in [6], has been developed under this approach.

Fault management and maintenance are vital aspects in industrial process. They not only include the knowledge about the failure modes and their causes, but an awareness of the extent to which equipment failure affects safety, product quality, costs and plant availability [8]. Changes in the maintenance approaches, recent knowledge, skills of maintenance people, and so on, must be incorporated into the maintenance systems in order to attain high performances. In this sense, maintenance systems should be developed in an open framework permitting to take into account the changes in the maintenance paradigms. In the maintenance framework, the challenges are related to

- the selection of the appropriate techniques for fault detection, isolation and diagnosis;
- the building of the appropriate models for detection, isolation, diagnosis and prediction;
- the decision-making in the design of feasible preventive maintenance plans;
- the decision-making in the emergency situations.

In this work, an agent-based reference model for fault management system (FMS) is proposed, as a part of the automation model proposed in [6], and it is based on the generic framework proposed in [7]. This model outlines the management of the preventive maintenance plans permitting to perform detection, isolation and diagnosis tasks. This model takes into account the following tasks:

- local or global abnormal behaviour detection of the controlled process;
- fault isolation and diagnosis;
- fault causes identification;
- preventive and corrective maintenance task running;
- maintenance task planning;
- fault prediction models development.

The FMS objectives are achieved by the coordinated interaction of the agents. Reasoning aspects as well as adaptive and learning aspects are incorporated into the multi-agent behaviour, helping in the interpretation of available data from different sources and characteristics. This way, the agent-based FMS provides the assistance in the detection–diagnosis–decision process, as well as in the planning and running of maintenance tasks.

2. Theoretical aspects

2.1. Fault management and preventive maintenance

Maintenance activities have attained the first places into the production processes tasks. These activities should assure the accomplishment of the plant functions and, therefore, the mains tasks should be preventives tasks aiming to the detection, diagnosis and prediction (on-condition tasks), the checking activities on the hard time intervals (on-time tasks), follow by the overhauling tasks in order to restore the equipment functionality. In this sense, preventive maintenance treats with the running of the inspection task and service that have been scheduled at specific time, in order to keep the equipment's capabilities and avoid a functional failure.

In the developing of a maintenance program, it should be stated what tasks are needed to do and when it must be done. This is guided by the extensive knowledge about the fault mechanisms, the experiences and the information provided by the equipment designers. Thus, data and information efficiently permitting to drive the detection–diagnosis–prediction task should be available. On the other hand, a maintenance program must satisfies new expectations, to give attention on the new researches about the failure mechanism, as well as the incorporation of new techniques and concepts in the maintenance area [8]. Reliability and availability become key concepts: in automated environments, failures impact the service and product quality. Industrial and environmental securities also claim for the development of a maintenance program satisfying these requirements. All of these aspects include the supplementary expectation that it be reached at lower costs.

Taking into account the previous aspects, a maintenance program should consider the following items [8]:

(a) Mechanisms for the potential failure detection, based on the development of new technologies in preventive maintenance.
(b) Technologies for failure analysis that are supported by mechanism permitting to report the failures and analyse them (the failures' root).
(c) The information management that aims to the development of the information systems permitting to store the corrective maintenance record about the equipment, the failure analysis record (historic detection, diagnosis and prediction), the alarms, and the maintenance costs.
(d) The decision-making supporting systems using the available information in order to propose new maintenance program according to the current enterprise context.

At the last years, engineering managers have known all about the impact of the maintenance on the industries performance, making an effort to providing support in the development of preventive maintenance activities. In this sense, maintenance must be seen as an enterprise strategy in order to attain major profit by viewing operation maintenance altogether. As a consequence, mechanisms and systems must be developed in order to integrate maintenance strategies with the business operational aspects [9].

In [10], the role of preventive maintenance has been remarked as a foundation in developing an integrated maintenance strategy by considering planning, scheduling, technique training and resources (inventory/acquisition). However, in the practice, just a few percentages of enterprises effectively implant their preventive maintenance programs.

Preventive maintenance is a key strategy for applying reliability-centered maintenance and predictive maintenance techniques as tools for condition monitoring from indexes associated to their reliability [11] and on-condition maintenance tasks [8,12].

Computer systems are a fundamental aspect to successfully implant a preventive maintenance strategy, for providing the information support, needed in maintenance management, and interfaces permitting to propose the repair work-flow identified after apply the predictive maintenance tasks (decision-making support) [10]. Different aspects to be taken into account in software systems integration in maintenance management have been discussed in [9].

In order to develop integrated maintenance systems, different approaches have been developed [13]. From a commercial point of view, most of suppliers of computerized maintenance management systems offer products based on preventive, corrective, reliability-centered maintenance [14–18]. These systems are composed by modules such as: equipment data and history, work order, trouble calls (maintenance petition), spare parts, purchase requisitions and orders, and inventory; these modules aim to support the maintenance planning [12,14,19]. These solutions, in general, do not offer an integrated computerized solution but provides an aggregate solution based on client/server architectures, without take a care of the operating system and of the used databases. In this way, it is incipient the development of products integrating the maintenance management, diagnostic and fault detection tasks with the supervisory system, this integration allows to carry out, in a coherent way, the maintenance planning based on condition of the productive process.

In [20] an integrated architecture for monitoring, fault diagnosis and maintenance is proposed. This architecture is composed by four modules: data calibration, condition monitoring fault diagnosis and maintenance assistance, integrating different artificial intelligence techniques. In [21], a methodological approach for developing maintenance systems is given, by defining the outlines for the distributed maintenance systems. In that work a maintenance reference is defined, by considering the maintenance activity as a global domain, i.e., the maintenance domain. This domain is composed by other domains sharing the maintenance objectives such as strategy, management, supervision, monitoring, fault diagnosis–correction–compensation, decision, execution, quality and evaluation. This maintenance domain permits to develop the integrated control, maintenance and technical management concept into a systemic framework.

From a distributed point of view, the e-Maintenance concept born as an evolution of the concept and techniques of e-Service and Tele-Service, in order to integrate monitoring, performance assessment methods and technologies, [12,22–24]. By using the e-Maintenance concept it is possible to achieve the integration of distributed knowledge, information and data in order to perform the equipment analysis, diagnosis tasks and to propose the best actions to maintain the well-working at the adequate time intervals (decision-making support). In [23], information software and object-oriented technologies as Web-services and XML have been used in order to propose a framework for the development of e-Diagnostic and e-Maintenance permitting to achieve the automation of diagnosis processes and their integration with the maintenance information. By the other hand, in order to perform the integration of the distributed experted knowledge, a MAS-based framework is proposed in [22,24] for accomplishing the cooperation, coordination and negotiation process between the main actors that have this knowledge: (i) production management, (ii) control expert and (iii) maintenance expert.

2.2. Multi-agent systems

Multi-agents systems (MAS) theory can be viewed as an evolution of artificial intelligence, in order to achieve autonomous computational systems. Although the agent definition has been argued into the researchers community of the distributed artificial intelligence (DAI), it is accorded that the autonomy is the main characteristic describing an agent, the autonomy being the ability to accomplish a task and reach its objectives without human, or any other, assistance [25,26]. Particularly, each agent has got properties as autonomy, mobility, rationality, sociability and reactivity. Furthermore, they may have built-in reasoning mechanism permitting to propose intelligent solutions and to evolve by experiences. Thus, intelligent agents have attributes and methods, like the object-oriented programming, and also they have built-in beliefs, desires and intentions, which are linked with their environment and provide them of the states that determine their behaviour [25,26]. The most important agent's characteristics are

- *Autonomy.* Agents are autonomous as they perform their tasks without human or any other external assistance. Autonomy is the agent's ability to have an own behaviour and reacts to external stimulus, base on its internal state. Each agent receives and senses the environment signals, analysing them by using internal mechanisms.
- *Communication*. Agents perform conversations in order to communicate with other agents. The agent's ability to talk by using a language based on ontology is an important step for implementing conversation into computational environments. The ontology is a set of concepts, predicates, sequences, terms and relations between these elements and they are understood by the agents' society. Thus, each agent understands a speaking intervention from other agent and, as consequence, the internal state of the agent can be modified and it can react beginning other speak interventions. Indirect communication is another type of communication and it is not based on messages but on shared information and environment measures.
- *Reactivity.* Reactive agents have the ability to perform immediate actions at time of received signal from the environment. Reactivity in agents permits quick actions in real time and this action need not complex rules.
- *Intelligence*. Intelligence is the main ability associated to the agent concept [26,27]. In order to analyse and take an action

in autonomous manner, agent should incorporate intelligent mechanisms. Intelligent techniques like expert systems, neural networks, genetic algorithms, etc., are often used for incorporating intelligent behaviour. By the other hand, intelligent communications may be also performed by using artificial life techniques like ant colonies.
- *Mobility.* Agent has the ability to move its state and program code from a node toward other one in a distributed system [28,29]. This ability allows the distributed computation in order to use local resources while the information is processed.

2.2.1. The design methodology of MAS

MAS-CommonKADS is a general methodology for the agents modelling [30]. This methodology proposes a set of models arranging the needed items for describing the MAS; these models are: agents, tasks, communications, experience, coordination and organization (see Fig. 1). Each model is described by templates containing the attribute identification and their relations.

MAS-CommonKADS methodology has two phases: the *Conception and Analysis* phase [31,32], phase determining the system requirements associated to the problem. In this phase, the actors and cases of use are developed as well as the models above mentioned. The role performed by a person, a piece of software or another system is represented by an actor; a case of use is associated with the needed actions in order to produce a useful result for an actor. The *Design* phase, phase studying the requirements defined in the previous phase, and architecture is proposed in order to satisfy these requirements.

In [33] the MAS-CommonKADS methodology has been enhanced by incorporating new aspects for agent modelling. The enhanced methodology, called multi-agent systems-based integrated automation (MASINA), is a new approach for MAS modelling in industrial automation processes. This methodology redefines the coordination, communication and experience models; and it extends the agents and tasks models. This new modelling approach allows for the development of

- *Agent model*. This model describes the characteristics of the agents proposed for the problem resolution; this model is the bridge toward the other models [34].
- *Tasks model*. In order to attain the agent's objectives, the tasks that should be performed by the agents are described in this model.
- *Intelligence model*. This model identifies the reasoning and learning mechanisms; and also defines the strategic knowledge about the domain, tasks and experience, in order to accomplish the MAS objectives.
- *Communication model*. Interactions between the agents are described here. This model defines the conversations by defining the sent messages and the speaking interventions.
- *Coordination model*. This model describes the coordination scheme between the agents, as well as the strategies for conflict resolutions, direct and indirect communication mechanisms, meta-languages and ontology.

2.3. Agents-based intelligent distributed control systems

Agents technology are being widely used as modelling approach for control systems and industrial developments because of the decentralized nature of the problem in these areas [35] and the complexity of the manufacturing and business environments [36]. The use of agent technology in industrial environments has been proposed in the past decade and the role of the integrated monitoring–diagnosis–decision-making activities

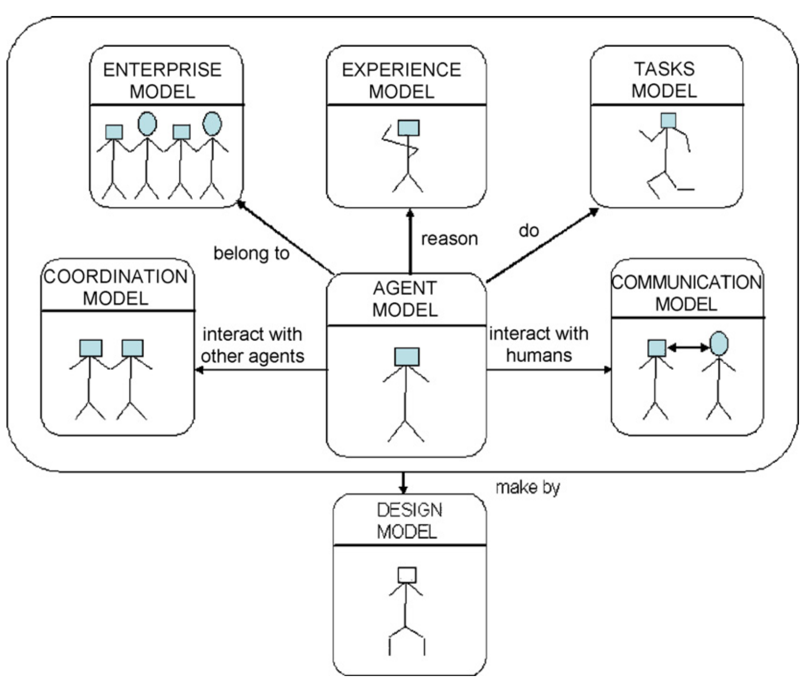


Fig. 1. The MAS-CommonKADS models.

have been highlighted in the approaches using agents. In order to integrate the multiples perspectives in the related areas in industrial environments (informatics, control, maintenance, management) and competing interests to attain global objectives, MAS-based architectures have been proposed [6,37–40] and the special attention in agent communication for industrial integration has been addressed in [39,41].

In [37], an intelligent agent framework integrating people and computer systems in complex, distributed manufacturing environments is proposed. Particularly, it highlights the necessity of automation and monitoring to achieve the enterprise integration in order to people and software has been warned about the abnormal event and then take an adequate decision. It is also introduced the necessity of knowledge availability about the process and equipment to be used in monitoring, diagnosis and decision-making activities. It proposes a generic approach for modelling these activities in cognoscenti and conceptual terms. In [38] the outlines for the development of agent-based models for manufacturing applications are given, remarking the existence of special roles of the agents providing the system operation.

Particularly, an agent is proposed for continuous monitoring detecting abnormal situation, taken the corrective actions or warning human operator about their occurrence. In [40] the use of agents in automation systems in their conceptual analysis is presented, remarking the role of the monitoring and maintenance as a part of the engineering process and this role is performed by an agent. In [6] a referential architecture based on MAS taking into account control, abnormal situation management, production planning, production resources management has been proposed and the functional analysis of the abnormal situation management has been addressed from a multi-agent point of view. Finally, recent works proposed in [22,24] show the important role of MAS-architectures in order to achieve e-Maintenance platforms.

Although the use of MAS-based approaches has been addressed in industrial environment and the importance of incorporating the supervision, monitoring and maintenance activities has been highlighted, no model of agents specification, from a methodological point of view, has been given in these approaches, particularly in the case of maintenance management problem.

In this work, the agent modelling is developed by using the AIDCS reference framework. The agents-based intelligent distributed control systems (AIDCS) is a MAS platform specifically designed for the industrial automation systems [7,33,34]. It proposes a collection of agents characterizing the elements of control of processes and defining a generic mechanism for the handling of the organized activities related with the industrial automation. This way, the agents of the AIDSC are

- *Measuring agent*. It gathers the necessary information in order to know the state of the process.
- *Controller agent*. It takes actions based on the system's state.
- *Coordinator agent*. It modifies the controller agent's decisions and establishes new objectives and services. It coordinates the agents' community.
- *Action agent*. It executes the decisions taken by the controllers, coordinators and/or specialized agents.
- *Specialized Agents*. They execute the special tasks of the agents' community.

The AIDCS is divided in two levels: an interaction level with the environment where the measuring and action agents live; and a decision level where the other agents of the community live (see Fig. 2). We call this set of five agents, the control agents community.

The AIDCS also proposes a community of agents that manages the control agents. This community is called the

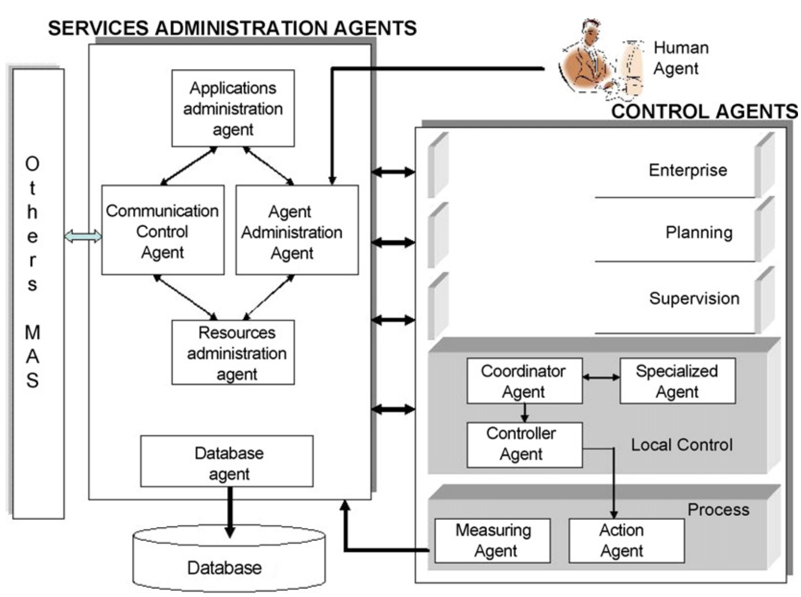


Fig. 2. AIDCS model.

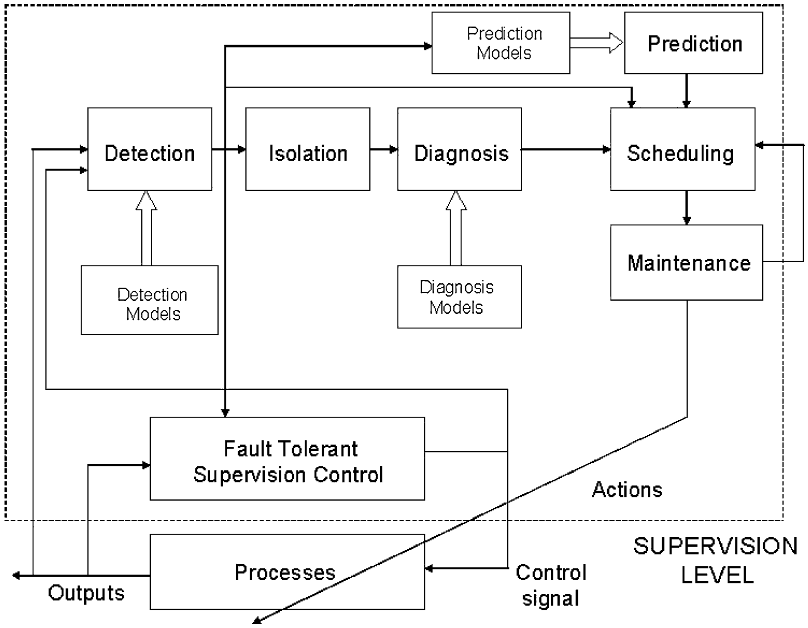


Fig. 3. Reference model for fault management.

services administration system (SAS) [34]. The SAS allows the migration, the localization, the activation/inactivation of agents, and the knowledge of the global state of the AIDCS; all these works are carried out by the agents of agents administration. Another work that the SAS performs is to control the inventory of all the applications and resources that are managed and/or are provided by the agents, these works are performed by the agents of applications administration and the agents of resources administration, respectively. On the other hand, it is also needed the administration of the data stored inside the system; this work is carried out by the database agent. Finally, the AIDCS has the capability to communicate with others SMA through the SAS, specifically by the communications control agent.

3. MAS-based reference model for FMS

In this section, a MAS-based reference model is specified from a methodological point of view, by using the five models mentioned in Section 2.2.1. The proposed model agrees the integrated concept of control, maintenance and technical management; the functionalities according to the maintenance reference domain are taken into account in the roles of the proposed agents. This MAS-based model provides a generic framework to solve the fault management because of the association of each maintenance activity with an agent service; the proposed model permits a comprehensive treatment of the fault management problem, it can be used in any area of industrial environments and it can be integrated with the supervision applications in order to support the decision-making on the controlled processes. Interactions between the proposed agents, defined in communication model and coordination model, permit an adequate information/decision flow in order to solve collaborative and distributed aspects.

3.1. Reference model for fault management systems

According to the aspects mentioned in Section 2.1, a reference model permitting the adequate interaction between the maintenance tasks is shown in Fig. 3 [42]. On this figure, it is remarked that preventive maintenance tasks are in the supervision level of control. Solid arrows denote the available information flow: the output variables of the process goes to the detection tasks; in case of fault detection, the information flow goes to the isolation, diagnosis and prediction tasks, each one is supported by models. Finally, the resulting information goes to the scheduling tasks in order to propose maintenance actions. Available information can be the current information or historical information about the detection, isolation, diagnosis and prediction and also about previous maintenance plans and corrective actions (feedback arrow between scheduling and maintenance).

The FMS proposed in this work is inspired on the previous reference model. Therefore, the FMS is composed by two blocks: the first block accomplishes monitoring and fault analysis tasks (MFAT); the second block accomplishes the maintenance management tasks (MMT). The FMS and engineering management are interacting blocks related to the productivity indexes, human and financial resources, components stock, and so on. On the other hand, the FMS also interacts with the fault tolerant controlled process, it being the final receiver of fault detection–diagnosis–decision tasks (see Fig. 4).

The MFAT block is concerned with the following tasks:

- *Detection* This task permits the identification of an invalid state in the process. A fault presence is stated from the behaviour of significant variables; as a consequence, base on this behaviour, the systems may be fault-free or in abrupt failure or incipient fault. Therefore, it is needed to have

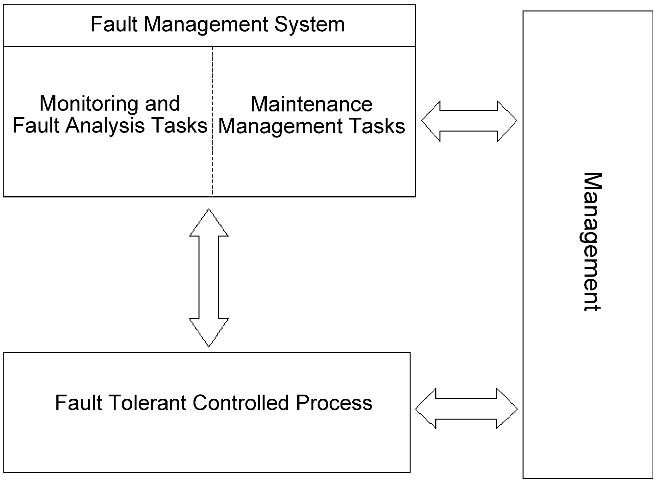


Fig. 4. Fault management system.

detection models and the information from the process is used in the detection.

- *Isolation* This task estimates the place where is occurring the faulty item.
- *Diagnosis* This task determines the failure mode related with the fault detection, their causes and consequences. Therefore, it is needed to have diagnosis models and the information from detection and isolation tasks is used in the diagnosis.

The MMT block is related with the following tasks:

- *Prediction* This task tries to estimate when an incipient fault goes to a functional failure; therefore, it is needed to have predictive models and the information from fault detection and diagnosis tasks may be used in order to build these models. Prediction tasks should estimate the failure occurrence and thus, it should avoid the corrective actions. In this sense, prediction result is an asset for the planning task.
- *Planning* This task provides a preventive maintenance plan proposing the maintenance tasks that avoid the occurrence of a functional failure. It must also propose a contingency plan in case of abrupt failure in order to avoid fatal consequences on the process. Resources feasibility related to a particular maintenance task must be reviewed before the plan is proposed (manpower and component costs, human resources, stock, etc.).

Finally, *maintenance action* is concerned to set-up and running the maintenance tasks according to the maintenance plan. These tasks are on-condition (detection–isolation–diagnosis) and on-time tasks. Maintenance actions are also related to set-up and running contingency plans in case of abrupt or unexpected failures.

3.2. MAS-based reference model for FMS

In this work, a MAS-based reference model is proposed for the problem of fault management in industrial processes. Taking into account the MAS-CommonKADS methodology, the conception and analysis phase is only developed here, and the MASINA methodology is used in order to propose the agents, tasks, communications, experience and coordination models.

The FMS is a system into the supervision level of the industrial processes. The FMS as MAS, should consider the following items:

- Information exchange between the levels of the industrial processes.
- Monitoring and analysis of variables from the lower levels of the industrial processes (local control).
- Fault detection, isolation and diagnosis mechanisms and reasoning mechanisms supporting the decision-making process in the preventive maintenance.
- Distributed processing: the FMS activities as fault detection/isolation and estimation of working index should be embedded into a distributed computational model.

3.2.1. Actors and case of use for the FMS

Each block described in Section 3.1, performs important roles for fault management. Table 1 identifies and describes the actors in the fault management problem and also, identifies the cases of use associated with each actor.

The cases of use are described by using the templates notation proposed by Rumbangh [43]. Table 2 shows the case of use ‘‘tasks scheduling’’ associated with the actor ‘‘scheduler’’ and the UML diagram is presented in Fig. 5. More details about the cases of use can be seen in [42].

Table 1
Actors and cases of uses of FMS

Actor	Description	Cases of use
Detector	It receives the information from the controlled process and identifies whether the system remains on the invalid state. It analyses if the invalid state is associated with an incipient or abrupt failure	System monitoring, state identification
Finder	It isolates the region where the failure occurs	Finding-failure
Diagnostician	It identifies the failure modes, their causes and consequences. It may also identify new failures modes, causes and consequences.	Failure analysis
Predictor	It estimates the time period that a potential or incipient failure goes to functional failure	Failure occurrence estimate
Scheduler	It schedules the preventive maintenance tasks, taking into account technical and financial available resources. It can also redefine the maintenance plans in case of a maintenance task is not set-up and running	Tasks scheduling, redefining plans
Executor	It runs the maintenance plan and proposes the corrective plans in case of emergency. It should notice whether the preventive or corrective tasks are not performed	Running tasks, reporting tasks

Table 2
Case of use "tasks scheduling"

Case of use Tasks scheduling
Abstract The actor *Scheduler* proposes a set of preventive maintenance tasks according to the technical and financial feasibility. This actor should set-up and running the timeline
Actors Scheduler, predictor, database and executor
Precondition Information about a potential or functional failure occurrence should be received from the actors detection, diagnostician and predictor
Exception Information is not received from the actors mentioned above. Information is not communicated to the actor executor

In the following, the different models are described according to the MASINA methodology, in order to get the MAS specification and requirements for the FMS.

3.2.2. Agent model

Agent identification is based on the actors that have been defined in the conception phase. According to the actors in Section 3.2.1, the FMS provides the following functionalities into the DCS:

- monitoring;
- detection, isolation and analysis of failures;
- predictions of failure occurrence;
- scheduling of preventive maintenance tasks;
- set-up and running of preventive plans and corrective actions.

These functionalities can be embedded into the agents defined in the generic conceptual framework for the agents-based intelligent distributed control systems (AIDCS) [34], which has been adapted to the fault management problem. In this sense, this problem is thought like a generic problem of feedback control system. The roles of the mentioned actors are allotted on the following agents:

- detector agent (specialized agent);
- finder agent (specialized agent);
- diagnostician agent (specialized agent);
- predictor agent (specialized agent);
- coordinator agent;
- controller agent;
- actuator agent;
- observer agent.

The *detector*, *finder*, *diagnostician* and *predictor* agents performs the actor's functionalities. The functionalities of the actors scheduler and executor are performed by the remaining agents. The *Coordinator* agent gathers the information about the process' items from the specialized agents and, based on this information, it schedules the maintenance tasks. The timeline should be defined according to the item's reliability and the failure effects. These aspects can suggest a decision-making process: take a corrective action or redefine the preventive maintenance plan; the last is also done if a maintenance task is not performed. This timeline can include a long time horizon (long-term (LT) plan) and not take into account the human resources and inventory on hand. In this sense, the *controller* agent takes the preliminary plan provided by the *coordinator* agent and it proposes a short time horizon plan (short-term (ST) plan) satisfying the human resources, inventory on hand and critical claims. That permits to give a daily or weekly plan for easy tracking. On-condition maintenance tasks are notified to the *coordinator* agent who calls to specialized agents; on-time and corrective maintenance tasks are notified to the *actuator* agent. The *observer* agent gathers the information about the process control in order to determine if a functional failure occurs; it also notices about the maintenance state and correlates it with the process state. Maintenance state is related with the foreseen maintenance tasks performance and process state is related with the

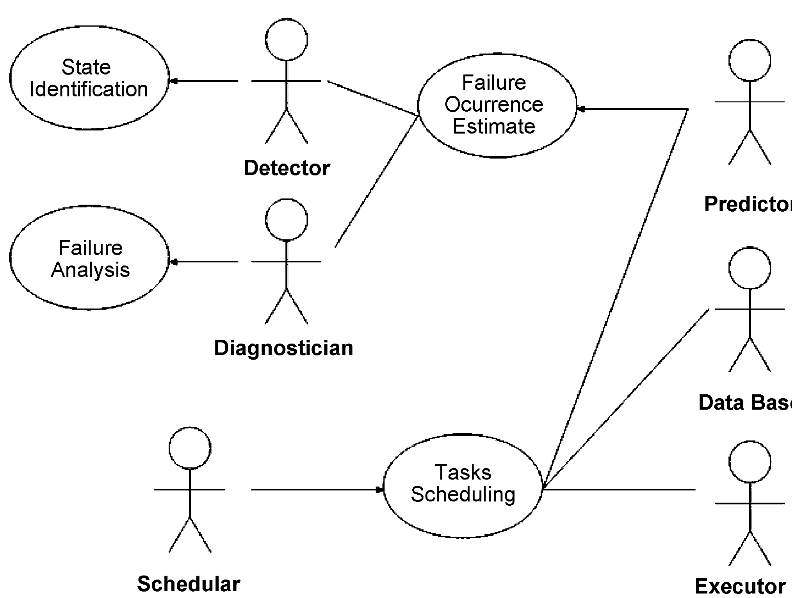


Fig. 5. UML use case "task scheduling".

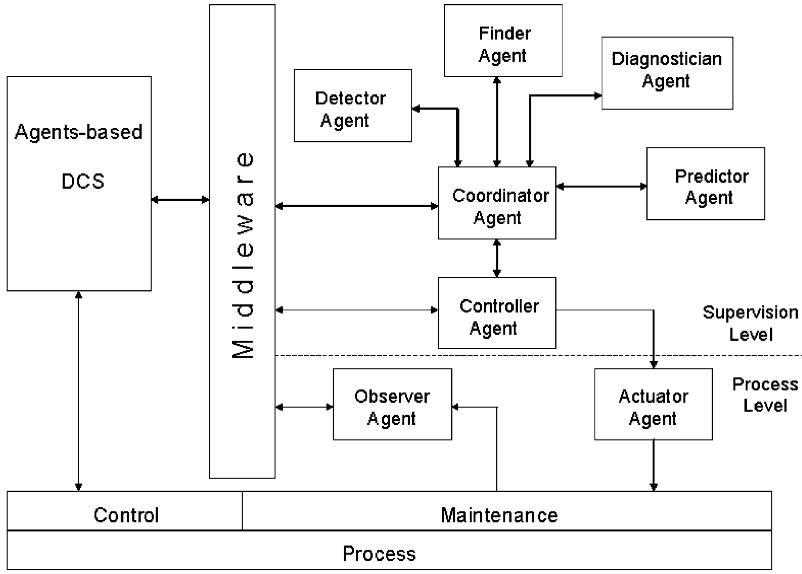


Fig. 6. The agents reference model.

operational function. The agent *actuator* performs the on-time maintenance tasks on the process.

Fig. 6 shows the MAS-based FMS who interacts with the AIDCS for the control process through the MAS-based services administration system (Middleware) proposed in [34]. The Maintenance Process is related with the whole maintenance engineering: available human, stock and financial resources and maintenance state.

Tables 3–9 show the characteristics and requirements of the coordinator agent: objectives, reasoning capabilities, abilities, constraints, provided services, etc. More details of the remaining agent are defined in [42].

Based on the main characteristics of the multi-agent systems, this agent-based reference model should support the different points of view or maintenance framework, this aspect being a part of the dynamical changes in the maintenance paradigms [8].

According to the functionalities and objectives of the coordinator agent, the following tables describe the services provided by this agent. The services should be able to propose or redefine an adequate maintenance plan permitting to reach high reliability on the process. It is also needed to have an adequate response time.

All services provided by the agents have common properties associated with:

- The service acquisition: it is related to the agent's availability. The agents are supposed always available and the services are performed.
- The service complexity: a service can be very complex according to the process' items.
- The service quality: it depends on the data reliability and the used method for obtaining the results.

Table 3
Coordinator agent

Agent Coordinator
Type Software agent
Roles Decision-making for the maintenance tasks planning. Coordination of the specialized agents and prioritization of urgent tasks
Description It gathers the information about the process' items from the specialized agents, and schedules the maintenance tasks. The timeline should be defined according to the item's reliability, the failure effects and urgent tasks. These aspects can suggest a decision-making: take a corrective action or redefine the preventive maintenance plan; the last is also done if a maintenance task is not performed

Table 4
Coordinator agent objective

Objective—Coordinator agent
To schedule the maintenance tasks according to the item's reliability, the failure effects and urgent tasks
Type On-condition objective
Input parameters Data from the specialized, observer and controller agents
Output parameters Maintenance plan and/or corrective maintenance order (urgent tasks)
Activation condition Information is received from specialized agents, observer agent, and controller agent
End condition The maintenance plan or corrective actions are proposed
Success condition A maintenance plan or corrective actions, permitting to achieve adequate reliability levels on the process' items, are proposed
Failure condition Success condition
Representation language Natural language
Description Coordinator agent assesses the different settings in order to create the best general maintenance plan on a long time horizon (long-term plan (LTP)) and produce the best set of corrective actions (urgent tasks) in case of emergency

Table 5
Coordinator agent service

Service—Coordinator agent
Name Maintenance plan proposition
Type Free, concurrent
Input parameters Data from specialized and observer agents
Output parameters Long-term maintenance plan
Representation language Natural language

Table 6
Coordinator agent service

Service—Coordinator agent
Name Maintenance plan redefinition
Type Free, concurrent
Input parameters Active flag about unrealized maintenance tasks (expecting tasks), active flag about functional or abrupt failures (urgent tasks)
Output parameters New maintenance plan, corrective maintenance orders
Representation language Natural language

Table 7
Coordinator agent service

Service—Coordinator agent
Name Specialized agent call
Type Free, concurrent
Input parameters Active flag about the application of detection, isolation, diagnosis and prediction tasks (DIDP tasks) on an process' items
Output parameters New maintenance tasks
Representation language Natural language

3.2.3. Tasks model

The tasks identification is done from the actors and cases of use defined in Section 3.2.1. The defined tasks for the MAS-based FMS are presented below:

1 Observer agent's tasks
  1.1 Abrupt o functional failures identification;
  1.2 Operation index estimation;
  1.3 Maintenance's state estimation.
2 Detector agent's tasks
  2.1 Statistical estimation about failures occurrence;
  2.2 Selection of the detection techniques;
  2.3 Incorporation of new detection methods.
3 Finder agent's tasks
  3.1 Finding failure.
4 Diagnostician agent's tasks
  4.1 Identification of failure modes and failure causes;
  4.2 Statistical estimation about failures modes;
  4.3 Statistical estimation about failures causes;
  4.4 Failure consequences analysis;
  4.5 Readjustment of diagnosis models;
  4.6 Incorporation of new diagnosis models;
  4.7 Identification of new failure modes;
  4.8 Identification of new failure causes.

Table 8
Coordinator agent capability

General capability—Coordinator agent
Ability To be able to propose or redefine an adequate maintenance plan permitting to reach high reliability levels on the industrial process
Representation language Natural language

Table 9
Coordinator agent constraints

Constraints—Coordinator agent
Type Temporal
Description Adequate response time is needful, specially for proposing corrective actions (urgent tasks)

Table 10
LT maintenance plan proposition

Task LT maintenance plan proposition
Objective Create a LT maintenance plan for each process' equipment
Precondition Information from the specialized agents must be clear and complete
Frequency It is related to the new information provides by the specialized agents
Description This task permits to create a LT maintenance plan, taking into account technical aspects, such as cycle of life of the equipment, prediction of functional failure occurrence from potential failure detection, critical consequences

Table 11
Application order of DIDP tasks

Task Application order of DIDP tasks
Objective Invoke the application of DIDP tasks
Precondition Identification of a functional failure occurrence. Request of on-condition maintenance tasks from ST maintenance plan. Potential failure detection
Frequency It is related to the preconditions activation
Description The coordinator agent manages the activities of specialized agents, in order to apply the DIDP tasks, if it is need

5 Predictor agent's tasks
  5.1 Reliability curve estimation of the process' items;
  5.2 Reliability index estimation of the process;
  5.3 Incorporation of new prediction models.
6 Controller agent's tasks
  6.1 ST maintenance plan proposition;
  6.2 ST maintenance plan processing.
7 Coordinator agent's tasks
  7.1 LT maintenance plan proposition;
  7.2 Resources evaluation;
  7.3 Application order of DIDP tasks;
  7.4 Application order of corrective action;
  7.5 Redefining the maintenance plans.
8 Actuator agent's tasks
  8.1 On-time maintenance task running;
  8.2 Urgent task running.

The following tables describe two of the coordinator agent's task. The remaining tasks are defined in [42] (Tables 10 and 11).

3.2.4. Intelligence model

At the beginning, the agents of FMS are reactive, that is, they have a set of initial believes and knowledge, but through the time, intelligent abilities may be incorporated. Out of the actuator agent, all agents in the FMS may be sensitive to be intelligent. A general structure of the intelligence model is presented in the following Tables 12–14.

3.2.5. Coordination model

This model describes the communications scheme of the MAS: conversations, protocols and associated languages. UML-based diagrams are used for describing the conversations. A set of six conversations has been defined for the fault management problem, each one with speaking interventions that will be described in the communication model:

Table 12
Agent's experience

Experience	
Representation	Rules
Type	Based on cases
Reliability	It is related to the data completeness
Processing scheme	Knowledge parameters tuning and new models incorporation

Table 13
Agent's learning mechanism

Learning mechanism	
Type	Adaptive
Representation	Rules, neural techniques, genetic techniques
Learning sources	Success or failures during the DIDP tasks
Update mechanism	Experiences feedback

Table 14
Agent's reasoning mechanism

Reasoning mechanism	
Information source	Previous results from de FMS's agents
Activation source	Scheduling tasks, DIDP tasks
Type of inference	Based on rules
Task–objective relationship	It decides if the used algorithm is adequate for the DIDP tasks or if an adequate maintenance plan is proposed
Reasoning strategy	It can be deductive or inductive: evaluate the causes of success or failure in the DIDP tasks or confront unknown situations in order to get rich the experience

(1) on-condition maintenance;
(2) maintenance plan;
(3) urgent maintenance tasks;
(4) redefining maintenance plan;
(5) maintenance state;
(6) functional failure identification.

In the following, the conversation *redefining maintenance plan* is presented. The remaining conversations are described in [42] (Table 15). Table 16 shows the associated coordination scheme, planning and communication mechanism for this conversation (Fig. 7).

3.2.6. Communication model

This model describes the speaking interactions used by the conversations. Each interaction between two agents is performed by sending a message. Taking into account the agents and tasks in the FMS, a set of 21 speaking interactions has been defined and they are suitably arranged into the conversations in the coordination model. In the case of the previous conversation *redefining maintenance plan*, the following speaking interactions are performed (Tables 17–19).

4. MAS-based FMS prototype design

The proposed MAS-based FMS has been developed on Java Development Framework (JADE) [44], a tool according to the Foundation for Intelligent Physical Agents (FIPA) request for the development of MAS applications. This tool supports the development of ontologies and codification schemes. JADE supports the remote methods invocation (RMI), common object request broker architecture internet inter-ORB protocol (CORBA IIOP) and events notification protocols; it supports the complex planning of agents' tasks, as well as the integration with the reasoning machine Java Expert System Shell (JESS). The source code is available under Lesser General Public License (LGPL) [45].

In order to develop the FMS prototype, four additional agents have been defined: main agent (MA), run agent (MA), monitoring agent (MA) and update agent (UA). These agents permit the implementation of the graphic interface of the FMS: built an agent, execute the MAS, visualize the performed agents' activities, and update the agent models in the system. For the agents programming, the FMS uses a system based on MAS for code generation (MASCoGen) [46]. The MASCoGen runs on JADE platform and permits the agents-oriented programming.

Table 15
Conversation description

Conversation	Redefining maintenance plan
Objective	To redefine the execution timeline of the maintenance tasks that have not been set-up and running on the process
Agents	Coordinator, database (MAS-based Middleware), human
Beginner	Coordinator agent
Speaking interactions	Expected maintenance tasks search, alarm, maintenance plan sending
Precondition	A particular maintenance task has not been set-up and running on the process
End condition	A new timeline has been proposed and a new plan is sent to the database agent, else, an alarm is sent to the human agent (user)
Description	This conversation permits the information search on the database about the maintenance tasks that have not set-up and running (expected tasks). The coordinator agent should analyse the information describing the particular tasks and it should take a decision about the run date. As a consequence, a new maintenance plan is proposed and it is available for the controller agent. If a new date is not provided, then, an alarm is sent to the user (see Fig. 7)

Table 16
Coordination scheme of the conversation *redefining maintenance plan*

Coordination scheme	
Objective	To coordinate the interactions between the Coordinator, database and Human agents
Type	Predefined
Coordination by default	Centralized, by using message pass between the agents
Planning	
Type	Predefined
Technique	Protocols are used for consulting and get information between the agents
Communication mechanism	
Type	Direct
Technique	Message pass
Meta-language	KQML
Ontology	It is related to the vocabulary about the fault management problem

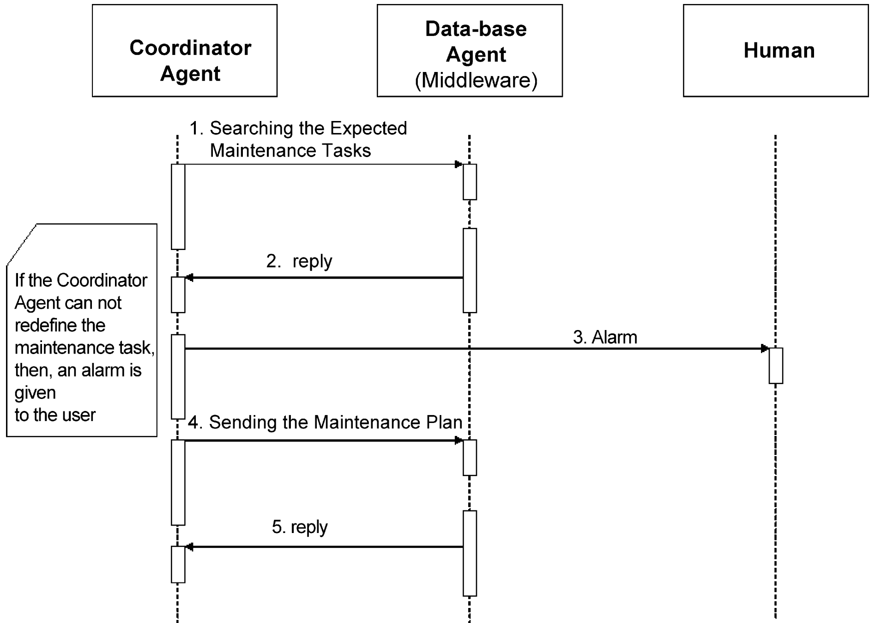


Fig. 7. UML interaction diagram of the conversation *redefining maintenance plan*.

The main agent interface (Fig. 8), is a single window showing three options: file, agents, and tools. File option has the exit option for finishing the application. Agents option has three options: built agents, run agents and agents monitoring. Built agents option permits to build the FMS agents (see Fig. 9), Run agents option permits to execute the FMS agents (see Fig. 10), and monitoring agents option permits to see the activities performed by an agent (see Figs. 11–15). Finally, tools option has the customize models option that permit update and modify the available agent's models in the MAS.

Table 17
Speaking interaction expected maintenance tasks search

Speaking interaction Expected maintenance tasks search
- *Type* Query
- *Objective* To search in the database the expected maintenance tasks that have not executed on the process
- *Agents* Coordinator, database (MAS-based Middleware)
- *Beginner* Coordinator agent
- *Precondition* An active flag about expecting tasks
- *End condition* The coordinator agent receives from the database agent, the whole information about the expecting tasks
- *Conversations* Maintenance plan redefinition, urgent maintenance tasks
- *Description* The coordinator agent requests the whole information about the expecting tasks reported by the observer agent

Table 18
Speaking interaction *alarm*

Speaking interaction Alarm
- *Type* Report
- *Objective* To warn about the unrealized agent's task
- *Agents* Coordinator, human (user)
- *Beginner* Coordinator agent
- *Conversations* On condition maintenance, maintenance plan redefinition, urgent maintenance tasks
- *Precondition* The particular task of the MAS is not performed
- *End condition* The human agent receives an active flag about an expecting task of the MAS
- *Description* The coordinator agent send an active flag about the unrealized task, in this case the task *maintenance plan redefinition*

Table 19
Speaking interaction maintenance plan sending

Speaking interaction Maintenance plan sending
- *Type* Information report
- *Objective* To save in the database the new maintenance plan
- *Agents* Coordinator, database (MAS-based Middleware)
- *Beginner* Coordinator agent
- *Conversations* Redefining maintenance plan, on condition maintenance
- *Precondition* A new maintenance plan has been proposed because of a potential fault detection, fault prediction, urgent maintenance tasks apparition or expecting maintenance tasks
- *End condition* The Coordinator agent send to the database agent, the whole information about the new maintenance plan
- *Description* The coordinator agent reports the new maintenance plan and this plan is saved by the database agent

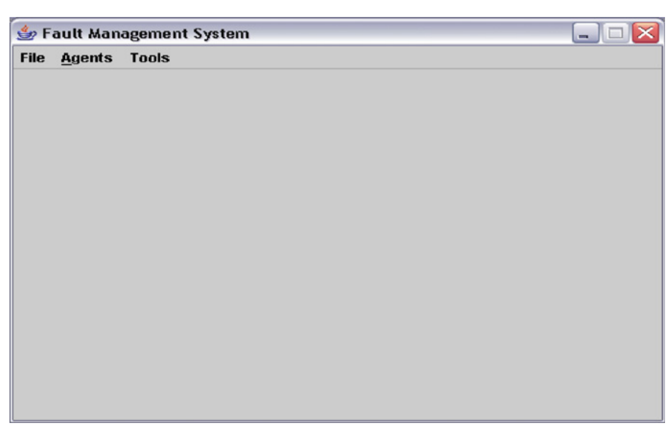


Fig. 8. FMS graphic interface: "main agent".

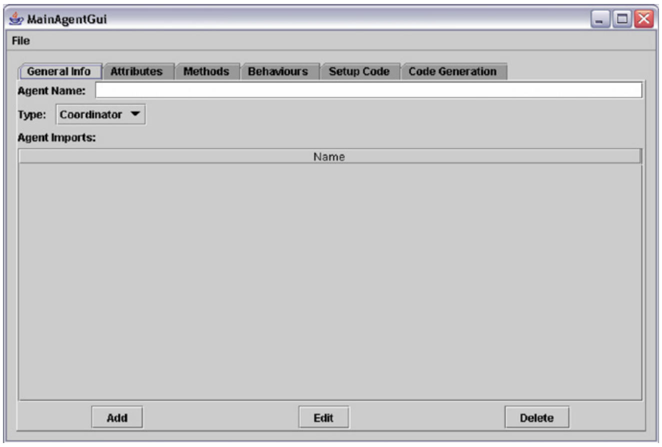


Fig. 9. FMS graphic interface: code generation by using the option "built agents".

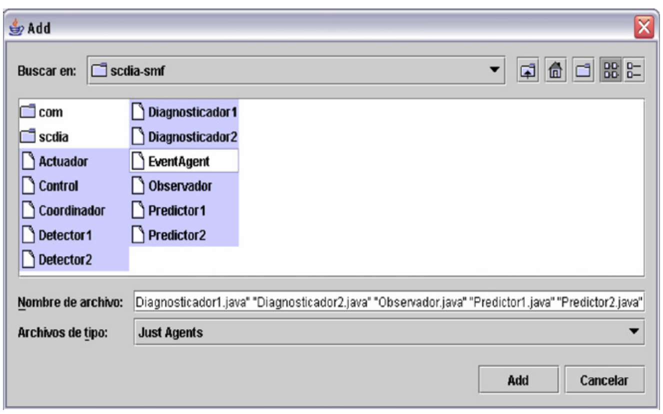


Fig. 10. FMS graphic interface: add an agent by using the option "run agents".

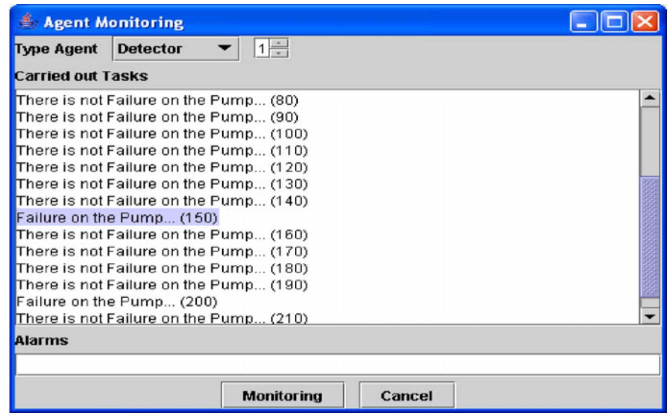


Fig. 12. FMS graphic interface: option "monitoring agents" on detector agent.

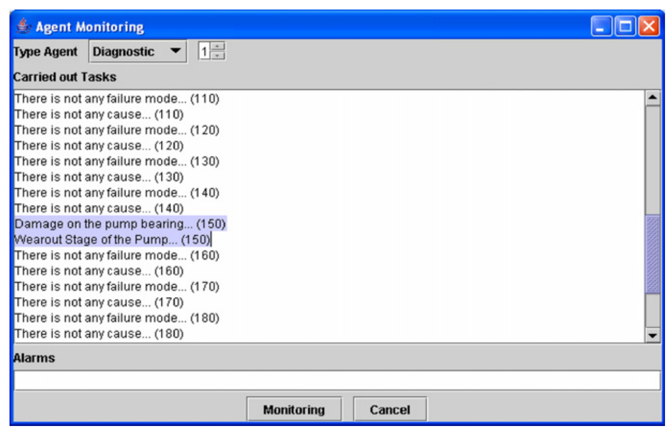


Fig. 13. FMS graphic interface: option "monitoring agents" on diagnostician agent.

4.1. Case of study: the pool pumping system

In order to validate the prototype performance, the maintenance activities on a pool pumping system has been studied [3]. The system function is to maintain the water flow, from the pool to the conditioning and heating systems, on top of 7 GPM (gallons per minute, 1 gallon = 3.785 l). The pumping system has the following equipment: main pump, main filter, pipelines, valves, electromechanical timers and other filters.

Only the main pump and the main filter have been considered in this work. In the operational environment, the pressure of the water supply must be considered and it can be decreased by the not well working of local aqueduct. In the test phase, normal and abnormal operational environments have

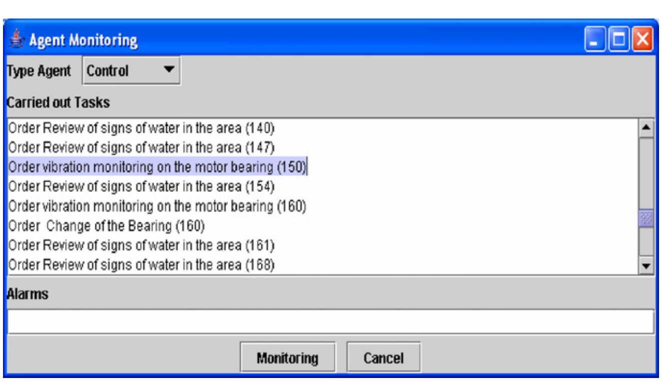


Fig. 11. FMS graphic interface: option "monitoring agents" on controller agent.

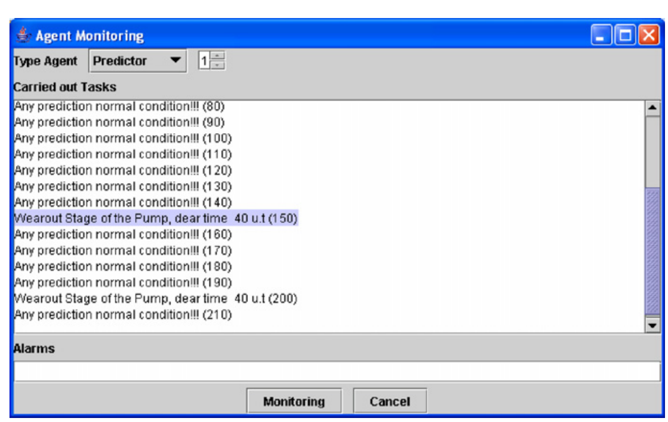


Fig. 14. FMS graphic interface: option "monitoring agents" on predictor agent.

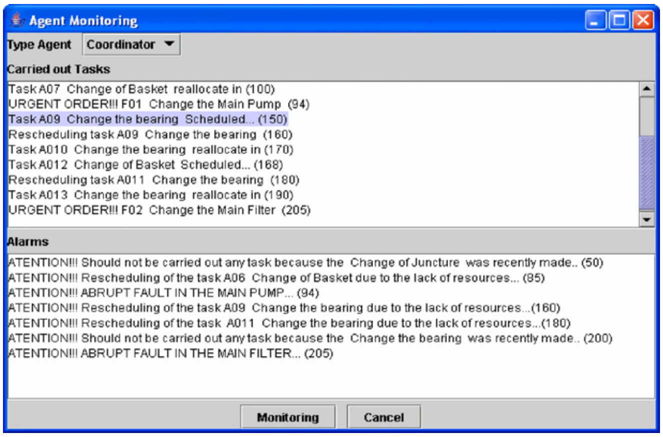


Fig. 15. FMS graphic interface: option ‘‘monitoring agents’’ on coordinator agent.

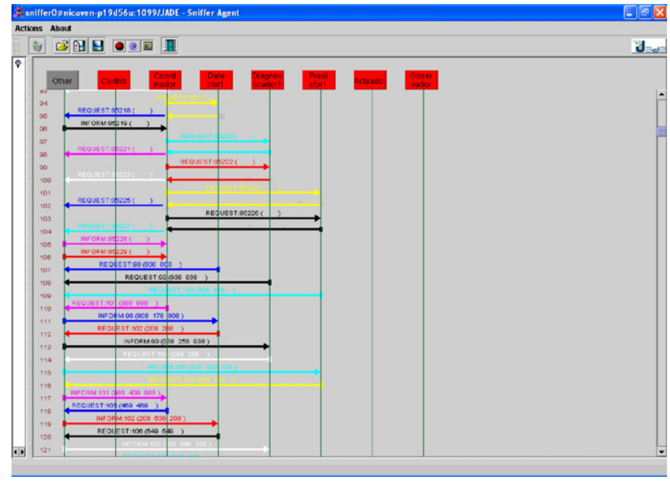


Fig. 16. JADE graphic interface: conversation *on-condition maintenance*.

been supposed. Failures modes, causes and consequences have been taken from [3]. In case of the main pump, the failure mode is the damage on the motor bearing, the failure cause should be the life cycle (wearout stage) and the critical variable associated to the failure mode is the vibration signal. A typical maintenance tasks associated to this failure mode are the vibrations monitoring (on-condition tasks) and the change of bearing (on-time tasks). More details can be revised on [3].

The proposed MAS-based FMS system in Fig. 6 has been defined by using the generation code tool in the prototype. Detector, finder and diagnostician agents have been created for the equipment. Observer agent and actuator agent have been created for the pumping system.

At the beginning, an initial maintenance plan has been proposed and a set of events has been defined on the normal and abnormal environment context, at different time points. A particular case during the validation phase is presented: a failure occurrence on the main pump has been noticed during the application of an on-time maintenance task. In this case, the performed conversation is ‘‘on-condition maintenance’’ and the FMS working is presented on the following figures.

The MAS is started by using the option ‘‘run agents’’ on the graphics interface in Fig. 10. The maintenance plan is known by the controller agent and the maintenance tasks are ordered at a particular time.

Fig. 11 shows the graphic interface of the agent monitoring: the controller agent orders the application of different maintenance tasks on the system’s equipment. Numbers in parenthesis are related to the current time unit (weeks, months, etc.). In this case, the highlighted order is the ‘‘vibration monitoring on the motor bearing’’.

The order from controller agent is received by the coordinator agent who coordinates the application of detection/diagnosis/prediction tasks. Fig. 12 shows the graphic interface in monitoring detector agent on main pump. In this case, the specialized detector agent runs the detection mechanisms and a fault on the pump is detected at 150 time units (the highlighted task on Fig. 12); at previous and later time points no fault is detected.

Once a fault is detected, coordinator agent requests information about the failure mode and causes to diagnostician agent. In this case, the diagnostician’s result is the damage on the pump bearing and the cause is the aging and wearout stage. Fig. 13 shows the Diagnostician monitoring and the highlighted results.

Then, coordinator agent needs more information in order to take a decision, then, the services of predictor agent are requested. In this case, the dead time for total failure is predicted at 40 after the current time, as it is showed on Fig. 14. Given the whole information, coordinator agent orders the change of the pump bearing at this time (see the highlighted order on Fig. 15) and this order is sent to the actuator agent. Fig. 15 shows the monitoring of coordinator agent. If a maintenance task is re-scheduled by the coordinator agent (carried out tasks) then this is shown on the alarm screen, and a new maintenance plan is stored.

All the conversations of this example are presented on Fig. 16. It shows de interaction diagrams and the speaking interactions between the agents in order to attain the global objective: execute and re-schedule the maintenance plan containing preventive maintenance tasks and run other tasks as diagnosis/prediction when necessary.

5. Conclusions

In this work, the conception and analysis of a multi-agents system-based reference model for fault management system has been developed, permitting to obtain an agents specification model, from a methodological point of view. This model agrees the integrated concept of control, maintenance and technical management; the functionalities, according to the maintenance reference domain, are taken into account in the roles of the proposed agents.

This MAS-based model provides a comprehensive treatment of the fault management problem, it can be used in any level of industrial environments and it can be integrated with the supervision applications without modifying the current automation architecture. Because of the services-oriented framework provided by the proposed generic agent model, software applications accomplishing the defined services can be integrated into the MAS model by using integration techniques. From the obtained specification models, the design model for implementing the proposed MAS-based fault management system can be developed in a natural way.

This model has been developed into a generic framework proposed for intelligent distributed control systems, thus, the fault management problem has been thought like a feedback control process. In this sense, the system performs a set of tasks (actions) permitting the maintenance tasks planning and the application of specific maintenance tasks as fault detection, isolation, diagnosis and prediction.

The MAS-CommonKADS methodology and the extended methodology MASINA provide a set of models permitting to describe the main characteristics of the MAS. The resulting models have a generic structure that permits to incorporate it into the automation process of any distributed control systems.

The agent's model permits to describe the main characteristics of the agents, that is their objectives, services, general capabilities and constraints. The whole functionalities are defined by the task model, and the manners in which the agents perform their tasks are defined in the coordination model. In the case of fault management systems proposed in this work, a set of six conversations has been sufficient to accomplish the FMS objectives; however, a set of 21 speaking interactions has been necessaries in order to develop these conversations.

The proposed MAS-based FMS has been developed on JADE and for the agents programming, the FMS uses a system based on MAS for code generation. A graphic interface permitting the implementation of the FMS has been developed in order to built an agent, execute the MAS, visualize the performed agents activities, and update the agent models in the system. A case of study related to the pool pumping system has been presented and the working of the FMS is showed, by using an event generator. In this case, the performance of the FMS has been validated, specifically the coordination and communication model.

Future works aims to the design of decision-making process in coordinator agent and the development of intelligence model.

Acknowledgement

This work has been supported by the grant no. I-621-98-02-A, Research Program of CDCHT-University of Los Andes. The authors would like to acknowledge this support.

References

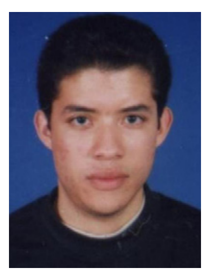
[1] T.J. Williams, P. Bernous, J. Brosvic, D. Chen, G. Doumeingts, Architectures for integrating manufacturing activities and enterprises, Computer in Industry 24 (1994) 11–39.
[2] J. Aguilar, M. Cerrada, K. Morillo, Intelligent hybrid systems: a reliability-based failure management application, Journal of Advanced Computational Intelligence 5 (6) (2001) 307–314.
[3] J. Aguilar, M. Cerrada, Un Sistema Clasificador Difuso para el Manejo de Fallas. *Revista Técnica de la Facultad de Ingeniería*, Universidad del Zulia 23 (2) (2000) 98–108.
[4] J. Aguilar, M. Cerrada, K. Morillo, A reliability-based failure management application using intelligent hybrid systems, in: A. Edelmayer (Ed.), Proceedings of the 4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes—SAFEPROCESS, vol. 1, Budapest, Hungary, (2000), pp. 197–202.
[5] M. Cerrada, J. Aguilar, Fuzzy classifier system: an application for fault tolerance in industrial processes, in: M. Hamza (Ed.), Proceedings of the IASTED International Conference on Artificial Intelligence and Soft Computing, Cancún, Mexico, (1998), pp. 453–456.
[6] C. Bravo, J. Aguilar, F. Rivas, M. Cerrada, Diseño de una Arquitectura de Automatización Industrial Basada en Sistemas Multiagentes, in: Proceedings of the IV Congreso de Automatización y Control CAC'2003, Universidad de Los Andes, Mérida, Venezuela, 2003.
[7] J. Aguilar, M. Cerrada, F. Hidrobo, G. Mousalli, F. Rivas, Application of the agent reference model for intelligent distributed control system, in: N. Mastorakis, L. Pecorelli-Peres (Eds.), Advances in System Sciences: Measurements, Circuits and Control, WSES Press, 2001, pp. 204–210.
[8] J. Moubray, Reliablity-Centered Maintenance, Industrial Press, 1992.
[9] I. Wray, Integrating EAM and PdM for maintenance productivity, Plant Engineering 57 (10) (2003) 65–66.
[10] T. Wireman, Managing predictive maintenance, Plant Engineering 57 (10) (2003) 58–60.
[11] I. Ushakov, R. Harrison, Handbook of Reliability Engineering, John Wiley, New York, 1994.
[12] Industry/University Cooperative Research Center for Intelligent Maintenance Systems, http://www.imscenter.net/.
[13] I.S. Kim, Computerized systems for on-line management of failures: a state-of-the-art discusión of alarm systems and diagnostic systems applied in the nuclear industry, Reliability Engineering and System Safety 44 (1994) 279–295.
[14] Maintenance Technology Suppliers, http://www.mt-online.com/suppliers/index.cfm.
[15] BENCHMATE, http://www.benchmate.com.
[16] CYBERMETRICS, http://www.cybermetrics.com/.
[17] SPL, Powering Strategic Enterprises, http://www.splwg.com/.
[18] AVANTIS, www.avantis.net.
[19] Maintenance Technology, http://www.mt-online.com/suppliers/index.cfm.
[20] M. Rao, H. Yang, H. Yang, Integrated distributed intelligent system architecture for incidents monitoring and diagnosis, Computer in Industry 37 (1998) 143–151.
[21] J.B. Leger, B. Iung, A. Ferro De Beca, J. Pinoteau, An innovative approach for new distributed maintenance system: application to hydro power plants of the REMAFEX project, Computer in Industry 38 (1999) 131–148.
[22] R. Yu, B. Iung, H. Panetto, A multi-agents based e-maintenance system with case-based reasoning decision support, Engineering Application of Artificial Intelligence 16 (2003) 321–333.
[23] M. Hung, K. Chen, R. Ho, F. Cheng, Development of an e-Diagnostics/Maintenance framework for semiconductor factories with security consideration, Advanced Engineering Informatics 17 (2003) 165–178.
[24] B. Iung, From remote maintenance to MAS-based e-maintenance of an industrial process, International Journal of Intelligent Manufacturing 14 (1) (2003) 59–82.
[25] G. Weiss, Multiagent Systems, The MIT Press, Massachusets, USA, 1999
[26] B. Waterbury, Artificial intelligence expands frontiers in asset management, Control Magazine (2002).
[27] J. Bigus, B. Jennifer, Constructing Intelligent Agents Using Java, 2nd ed., Wiley Computer Publishing, Canada, 2001.
[28] J. Nelson, Programming Mobile Objects with Java, John Wiley, 1999.
[29] B. Spell, Professional Java Programming, Wrox Press, 2001.
[30] C.A. Iglesias, M. Garijo, J. C González, J.R. Velazco, Analysis and design of multi-agent systems using MAS-CommonKADS. In: M.P. Singh, A.S. Rao, M. Wooldridge (Eds.), Intelligent Agents. Part IV. Agents Theories, Architectures and Language, Lecture Notes of Artificial Intelligence 1365 (1999) 1–16.

[31] I. Jacobson, M. Christerson, P. Jonsson, Overgaard. Object, Oriented Software Engineering. A Use Case Driven Approach, ACM Press, 1992.
[32] C. Potts, K. Takahashi, A. Anton, Inquiry-Based Scenario Analysis of System Requirements. Technical Report, GIT-CC-94/14, Georgia Institute of Technology, 1994.
[33] J. Aguilar, M. Cerrada, F. Hidrobo, F. Rivas, Especificación detallada de los agentes del SCDIA, *Technical Report*, Grant Project Agenda Petróleo no. 97003817, CEMISID-Universidad de Los Andes, Mérida, Venezuela, 2004.
[34] J. Aguilar, G. Mousalli, V. Bravo, H. Díaz, Agentes de Control y de Gestión de Servicio para el Modelo de Referencia SCDIA, in: Proceedings of the II Simposio Internacional de Automatización y Nuevas Tecnologías, TECNO2002, Mérida, Venezuela, (2002), pp. 45–50.
[35] N. Jennings, S. Bussmann, Agent-based control systems, IEEE Control Systems Magazine 23 (3) (2003) 61–73.
[36] V. Marik, D. McFarlane, Industrial Adoption of Agent-Based Technologies, IEEE Intelligent Systems 20 (1) (2005) 27–35.
[37] J. Pan, J. Tenenbaum, An intelligent agent framework for enterprise integration, IEEE Transactions on Systems, Man, and Cybernetics 21 (11) (1991) 1391–1408.
[38] M. Cantamesa, Agent-based modelling and management of manufacturing systems, Computers in Industry 34 (1997) 173–186.
[39] D. Ouelhadj, S. Petrovic, P.I. Cowling, A. Meisels, Inter-agent cooperation and communication for agent-based robust dynamic scheduling in steel production, Advanced Engineering Informatics 18 (2004) 161–172.
[40] T. Wagner, Applying agents for engineering of industrial automation systems. In: M. Schillo et al. (Eds.), Lecture Notes of Artificial Intelligence MATES 2831 (2003) 62–73.
[41] Y. Nahm, H. Ishikawa, A hybrid multi-agent system architecture for enterprise integration using computer networks, Robotics and Computer-Integrated Manufacturing 21 (2005) 217–234.
[42] M. Cerrada, J. Aguilar, J Cardillo, R. Faneite, Agent-based reference model for fault management, Technical Report, Grant Project I-621-98-02-A, Universidad de Los Andes, Mérida, Venezuela, 2002.
[43] J. Rumbangh, OMT: the development process, Journal of Object Oriented Programming 8 (2) (1995) 8–16.
[44] Jade-Java Agent Development Framework, http://jade.cselt.it.
[45] Jade-Java Agent Development Framework, http://jade.tilab.com/community-3rdpartysw.htm.
[46] W. Zayas, Diseño e Implementación de la Comunidad de Agentes de Generación de Código del SCDIA. MSc Thesis. Facultad de Ingeniería, Universidad de los Andes, Mérida, Venezuela, 2004.

Professor Mariela Cerrada was received the BS degree in System Engineering in 1993 and the MSc degree in Control Engineering in 1998, from the Universidad de Los Andes, and the PhD degree in Automatic Systems in 2003 from the Institute National des Sciences Appliqueés INSA-Toulouse-France. She is an Associate Professor in the Department of Control Systems at the Universidad de Los Andes and researcher of the Center of Studies in Microelectronics and Distributed Systems (CEMISID) and the Laboratory of Intelligent Systems (LabSIULA). She was the head of departament from 2004 to 2006. Her research includes the field of control systems, computational intelligence and automation. She has published more than 50 academic papers.

Professor Juan Cardillo was received the BS degree in System Engineering and the MSc degree in Control Engineering, from the Universidad de Los Andes, and the PhD degree in Automatic Systems in 2003 from the Université Paul Sabatier-Toulouse-France. He is an Associate Professor in the Department of Control Systems at the Universidad de los Andes and researcher of the Dinamical Systems Group and the Laboratory of Discrete Systems, Automation and Integration (LASDAI). His research includes the field of control systems, optimization, discrete-event systems and automation. She has published more than 60 academic papers.

Professor Jose Aguilar was received the BS degree in System Engineering in 1987 from the Universidad de los Andes-Merida-Venezuela, the M. Sc. degree in Computer Sciences in 1991 from the Universite Paul Sabatier-Toulouse-France, and the PhD degree in Computer Sciences in 1995 from the Universite Rene Descartes-Paris-France. He was a Postdoctoral Research Fellow in the Department of Computer Sciences at the University of Houston (1999–2000). He is a Titular Professor in the Department of Computer Science at the Universidad de los Andes, researcher of the High Performance Computing Center of Venezuela (CeCalCULA) and of the Center of Studies in Microelectronics and Distributed Systems (CEMISID). Currently, he is coordinator of this Center and head of the Science and Technology Bureau of the Merida State, Venezuela. His research includes the field of parallel and distributed systems, computational intelligence, combinatorial optimization, pattern recognition and control systems. He has published more than 190 papers in journals and five books in the domain of computational sciences, and science and technology management. He has received several awards and some of his papers have received special awards. The last 4 years has been one of the two best researchers of his university (it has more than 1000 researchers).



Raul Faneite was received the BS degree in System Engineering in 2002 and the MSc degree in Computer Science in 2004, from the Universidad de los Andes-Merida-Venezuela. He is systems analyst at the Venezuelan Petroleum Enterprise PDVSA-CRP. His research is related with the development of computational systems and programming.