DESIGN OF AN INDUSTRIAL AUTOMATION ARCHITECTURE BASED ON MULTI-AGENTS SYSTEMS

Cesar Bravo *, José Aguilar Castro **, Mariela Cerrada, Francklin Rivas Echeverría ^**

** PDVSA*
West Division. Industrial Automation.
+58-264-803750
bravocn@pdvsa.com
^ Universidad de los Andes
Postgrado en Ingeniería de Control y Automatización
+58-274-2402846
rivas@ula.ve
*** Universidad de los Andes*
CEMISID
{aguilar,cerradam}@ula.ve

Abstract: New paradigms in industrial automation aims towards intelligence distribution among components of the productive process and through the integration of several systems and applications in the companies, searching processes improvement, supporting decision-making tasks by real time information available, reliable and opportune. This work proposes industrial automation architecture based on multi-agents systems and uses the framework SCDIA to describe this architecture.

Keywords: Artificial Intelligence, Industrial Automation, Agents, Production Systems.

1. INTRODUCTION

Theory of multi-agents systems seems to agree with the tendency of industrial world, especially in oil & gas industry, distributing intelligence among the components of the productive process, providing them with mechanisms that allow them being conscientious of their states, goals and actions, and therefore they can be able to do self-regulation, self-diagnosis and take actions to maintain their desired state. Multi-agents systems consist in an agents' collection, each one with a partial vision of their environment but with social characteristics that allow them to cooperate for reaching individuals or global goals.

This paper proposes a multi-agents system that represents the operations carried out in an industry, from a distributed control perspective. This system is modeled using the framework SCDIA (Intelligent Distributed Control System based on Agents, in English), which is a multi-agents system architecture, built specifically for automation and control domains (Aguilar et al., 2001, Bravo et. al, 2003). This model contemplates five agents that represent elements of a control loop, thus, on SCDIA will be agents like, controllers, observers, actuators, coordinators and specialized agents. Besides, SCDIA proposes Services Administration System (SAS), which have the responsibility to develop integration tasks between the SCDIA components and external systems.

The SCDIA uses principles of Multi-Agent Systems paradigm for modeling industrial automation processes under a generic and reusable schema, useful for process control, resource administration and decisions making activities.

This paper is composed by four parts. The first part proposes a multi-agents model for an industrial automation platform design. The second one presents a case of study apply to oil production industry. In the third section is presented the design of agents related to the case of study already mentioned. The last part presents the conclusions of this investigation.

2. INDUSTRIAL AUTOMATION ARCHITECTURE BASED ON MULTI-AGENTS SYSTEMS

2.1 Generalities.

The automation architecture proposed in this paper is composed by three abstraction levels, each one represented by a Multi-Agent System (MAS).
In a first level the productive process is modeled as a MAS, where every production unit is represented as an agent.
In a second level, each agent of the first one is seen as a MAS where its components are responsible of the required activities to reach the goals of each production unit. Due to these activities are common for each production unit, all the agents at the first level will have a similar architecture.
Finally, since agents activities at second level are complex, is proposed a third level of abstraction where required tasks and calculations in each second level activity are developed by specific agents. As first level, agents at second level will be seen as MAS.

2.2 First Level: Business Objects Representation.

This level looks for represent production chain elements over company's TI platform, considering its characteristics, behavior and states in detailed, and the interrelation among them defined by the company's business rules. Representing production units as agents is based on the idea of giving intelligence and autonomy to each element in the productive process. .
The architecture of this level is metamorphic, that is, the quantity of agents in this level and their characteristics will depend on the process that is being modeled. In this way, our architecture can be adapted to any kind of production organization. The architecture's metamorphic characteristic at first level is referred only to which business objects will be modeled as agents, it's not referred to agent's internal architecture because this subject belongs to the second level.

2.3. Second Level: Functional Enterprise Processes Representation

On this level, the functions that any business objects must do to guarantee its production control are represented. The model proposed for this level is based on *Functional Data Flow Model,* described at ISA 95.00.01 Standard (ISA 95.00.01, 2000). Thus, each Business Object Agent is seen in this level as MAS, composed for the following agents:
Process Control Agent: this agent represents process control applications, from the field level, modeling local control applications executed in devices as Programmable Logic Controllers (PLC's) or Remote Terminal Units (RTU's), until supervisory level, modeling SCADA/DCS applications. This agent's architecture follows the SCDIA framework, since this framework has been designed exactly to model process control tasks.
Production Planning Agent: This agent defines the business object's production plan assigning production quotas and methods based on conditions, capacities and environment relation of this business object. Also, it emits production orders and establishes the production sequence. On the other hand, this agent defines production optimization methods.
Resource Management Agent: this agent manages resources (infrastructure, human resources, energy, etc) necessaries for business objects tasks execution in the productive process. Also, it manages inventory and final products delivery.
Maintenance Engineering Agent: this agent makes fault detection, diagnosis, prediction and isolation. and elaborates dynamical preventive and corrective maintenance plans; identifies faults that can be presented in a process, starting from analysis of state and historical information, providing a data model that contains all the information about this faults and the maintenance plans that should be executed to prevent or solve them in case of an abrupt fault.
Abnormal Situations Management Agent: this agent manages abnormal situations that can happen in the process, using the data model provided by the Maintenance Engineering Agent, and a knowledge base that collects the experience of operators, engineers and experts in process abnormal situations management. It is responsible abnormal situations detection, generate alarms and execute corrective actions. This agent is a MAS that has an architecture based on the SCDIA. Therefore, this MAS will be composed by the five agents of SCDIA, each one with specific tasks for abnormal situations management.
Under this scheme a common architecture for Business Objects Agents is proposed, based on standards in order to assure a generic and reusable model useful to any production unit to be modeled, and giving support to the first level metamorphic architecture.

2.4. Third Level: Calculation Agents Level

In this level is proposed modeling each one of the agents of the functional enterprise model level as MAS. Each second level agent can be modeled using a different framework, with architectures that will depend on the objectives of each one of them. Nevertheless, in this work its propose to use the SCDIA framework to model the agents of the second level, since it's a generic design that can be adapted to every production control activity.

To exemplify the third abstraction level agents modeling was chosen the Abnormal Situations Management Agent. Therefore, each one of the SCDIA agents is used, and its specific activities referred to abnormal situations are described next.

Observer Agent: this agent will have the mission of gathering the data coming from data repositories that can give information about the process state. Also, it can pre-process and/or validate the data, make observation of states and any other operation to obtain the information required by other agents to carry out abnormal situations diagnosis, predictions and treatment. This agent has available for any other MAS components the exact information about the process state.

Controller Agent: this agent compares current process state with the desire conditions. In case where current conditions are far away from a tolerance band, The agent executes control orders (stored inside an inference motor), which can be: alarms activation, diagnosis applications execution, characterization in conditions operation, etc., depending on current situation. This agent applies models of detection and diagnosis of abnormal situations to determine their causes and evaluates their consequences, using the data provided by maintenance engineering agent.

Actuator agent: depending on the decisions taken by controller agent, this agent could activate alarms and makes them visible for each actor involved with the problem resolution (SCADA/DCS operators, engineers of optimization, maintenance engineers, etc.). Besides this agent produces changes in SCADA/DCS (for example, it changes the set point in a control logic), executes workflows linked with maintenance plans to solve abnormal situations that are presented, including corrective maintenance measures.

Coordinator agent: this agent acts as a supervisor of SCDIA. It supervises system's inference motor operation, and modifies it if is necessary. Also it could modify workflows, change established values for normal operation conditions (for example, value nominal for process variables) and modify the inference motor structures of MAS components, through learning mechanisms based on artificial intelligence techniques. This agent executes tests that allow to identify and locate faults, establishes corrective maintenance plans (together with Engineering Maintenance Agent), and emits the services requirements to the specialized agents.

Specialized agents: in certain circumstances, inside fault detection and diagnosis process is necessary to carry out activities of data mining, mathematical and statistical calculations, prediction, etc. These activities are performed by specialized agents, each one of them with a specific task to accomplishment.

All agents of each abstraction levels are using the services provided by SAS. SAS guaranteed the communication among all agents and the efficient resources and services administration required.

The proposed architecture allows to model in a distributed way the applications that support the productive process, through the specifications defined in the second abstraction level.

Finally, the design of diverse abstraction levels allows approaching the modeling of complex systems through the decomposition of them in MASs.

3. CASE OF STUDY: OIL PRODUCTION PROCESS

In this paper, the case of study chosen was the oil production process, specifically, a Unit of Exploitation of Oil Reservoir (UEY, abbreviation in Spanish) model. It is proposed the design of five agents that represent the most important facilities of an artificial gas lifting oil exploitation loop in an UEY. These agents are:

Oil Well Agent: this agent is responsible for all the activities necessary for an oil well operation. Besides, this agent will have capacity to develop control and supervision, maintenance tasks programming and/or some repair activities in oil wells (like chemical injection), economical analysis, and other activities related to this business object. On the other hand, it will have the auto evaluation capacity and driving actions for the production methods optimization.

Flow Station Agent: it models the flow station operation. In addition, it controls and observes separators, pumps and other devices related to this facility. Also, it carries out the production planning based on the flow station operation and wells operation associated to it. Therefore, this agent will contain methods of optimization for the gas/oil separation and UEY productivity analysis .

Compressor Plant Agent: this agent observes and controls the gas compression activities in the exploitation loop and, through coordination with other agents, it will be able to plan the gas consumption of Artificial Gas Lifting Multiple (MLAG, abbreviation in Spanish) and other UEY facilities, and even, in external facilities.

MLAG Agent: this agent manages the injection gas distribution activities. Through this agent, it can be made the gas distribution planning in the UEY, and apply production optimization methods for the artificial gas lifting wells.

Tanks Farm Agent: This agent allows establishing optimization methods for the movement and supply of oil in a Tanks Farm, as well as oil pre-treatment.

The interrelation among business objects can be represented through a functional diagram. In this diagram, is presented functions performed in the oil production process.

The diagram on fig. 2 represents a general outline on how would be the interrelation among first level agents; a detailed model requires a deeper study of the functions that are developed in an UEY. Several colors were used inside the diagram, specifying which agent performed which function.

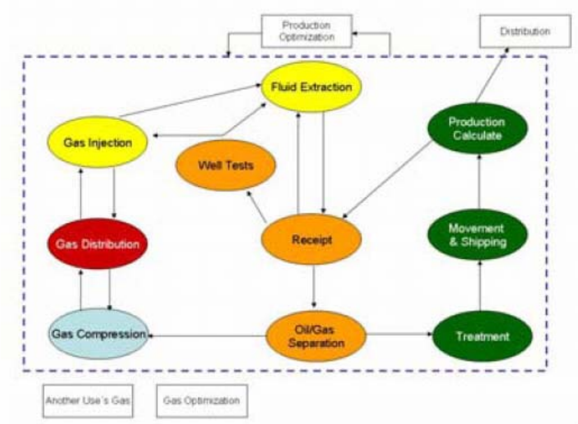

Fig. 1. Functional diagram of the First Abstraction Level.

The functions identified in the diagram are: *Oil Extraction*, performed by Oil Well Agent; receipt, measure of wells[1] and oil/gas separation, performed by the Flow Station Agent; gas compression and distribution, performed by Plant Compressor Agent; Injection of Gas, performed by MLAG Agent; and movement and supply, treatment and calculation of production, performed by Tanks Farm Agent..

Modeling the MAS behavior in the second abstraction level, it's used the *Functional Enterprise Control Model*, proposed on ANSI/ISA 95.00.01 standard, which describes the functions and information flow among production company's components.

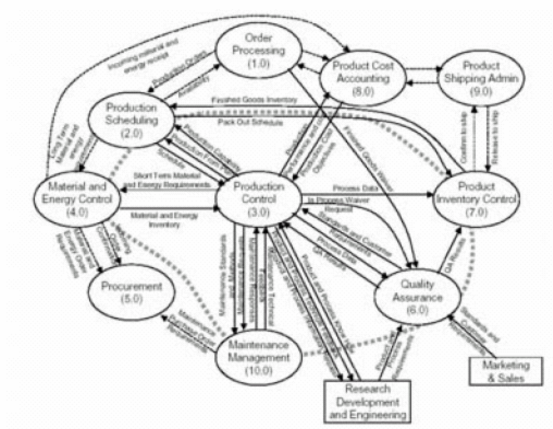

Fig. 2 Functional diagram of Second Abstraction Level[2]

ANSI/ISA 95.00.01 standard describes in detail each one of the functions specified in the fig. 2.

Now, it uses the Abnormal Situations Management Agent to explain the structure of the third abstraction level. To describe the MAS behavior that carries out the management of abnormal situations activity, will be used a Petri Net to represent the agent state transitions before an abnormal situation occurrence The fig. 4 shows this Petri Net.

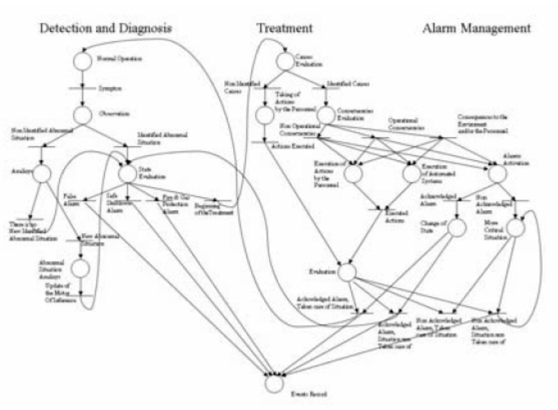

Fig. 4 ASMA Behavior before an abnormal situation

Petri Net presented on fig. 4 is divided in four parts: Detection, Diagnosis, Processing and Alarms Management.

The Detection and Diagnosis parts, performed the following tasks: Information gathering (carried out by the observer agent), Symptoms Detection (developed by observer and controller agents), Tests Execution (Developed by coordinator agent), abnormal situation Identification (carried out by controller and coordinator agents), Identification of new faults (maintenance engineering agent, at second level), Critical evaluation (carried out by controller agent), Safe Stop Alarm (developed by controller agent), Fire and Gas Protection Alarm (developed by controller agent), Events Log (developed by actuator agent).

Processing part performed the following tasks: Root Cause Identification (carried out by controller and coordinator agents), Consequences Evaluation (carried out by controller and coordinator agents), and Corrective Action Execution (carried out by actuator agent).

The alarm management part makes the following tasks: Alarms Emission (carried out by controller agent), Change state of alarms (carried out by observer and controller agents), and Process Evaluation (carried out by observer agent).

Negotiation processes that should be developed in the three abstraction levels proposed, are based on a coordination model of FIPA (Foundation for Intelligent Physical Agents) (www.fipa.org), where the communication is based mainly in message passing, using techniques as brokering, broadcasting, etc.

4. AGENTS DESIGN

Agents design was developed under FIPA's agent specification model, considering the existence of SAS, for MAS administration. The design presented provides the necessary tools for agents' installation in any MAS platform.

The agents name allows identifying each agent in the architecture, the domain to which its belongs and its related agents, considering the relationship among the

---

1 Well measurement is a procedure which measures the production characteristics of each well (gas percentage, oil percentage, API gravity, etc.). This procedure is located in flow stations.

2 This model is inspired on the ISA 95.00.01 Standard

three abstraction levels. Names are defined by SAS and it follows the FIPA agents' specification rules, with a modification that has been added which is an attribute that allows linking the agent with an agent in a superior level, that is, it allows to establish a hierarchy of agents according to abstraction levels described in previous sections. Therefore, each agent will have the following attributes:

ID (name@domain), it is composed by two parts, the name that identifies the agent, and the domain that indicates the site where it works. An agent can change it domain through a migration.

Addresses: is the network address where agent is located, for example iiop://domain1.com.

Services: services provided by agent;

Protocol: protocols manage by the agent. (Described in coordination model).

Ontology: sub-group of language that an agent dominates.

Language: language that agent uses to communicate with other agents.

Parent: agent in a superior level which is related to the agent. (This attribute was added especially for the architecture proposed in this work).

Similar to previous sections, the agents will be described by abstraction level.

Finally, it is necessary to explain that, since in the architecture proposed on this work there are links among agents of the different abstraction levels, when an agent is created, automatically are created the agents of the inferior levels that should be linked it (or if they already have been created, they are migrated to the agent's domain and they are assigned to him). Likewise, if an agent of a superior level is suspended or eliminated, the agents at inferior levels are also suspended while waiting for SAS destroys it, or links them with another agent (possibly include migration).

5. CONCLUSIONS

In this paper is presented an architecture for industrial automation platforms based on multi-agent systems.

The proposed architecture consists of three abstraction levels that allow modeling in a distributed way, the activities of industrial automation, giving to each business object autonomy and intelligence, and distributing tasks of each object among the three abstraction levels.

One of the main strengths of the proposed model is the metamorphic architecture of the first abstraction level which grants the flexibility of modeling any kind of production oriented company.

Second abstraction level agents, communication and coordination models of agents and their design are based on standards. These standards give to architecture reliability, robustness and scalability.

REFERENCES

Aguilar, José, Cerrada, Mariela et al. Aplicación de Sistemas Multiagentes a Problemas del Mundo Real. XXVII Conferencia Latinoamericana de Informática (CLEI). 2001.

Aguilar, José, Rivas Francklin et. al. Tercer Informe Técnico Proyecto Agenda Petróleo. Universidad de los Andes. Mérida. 2002

ANSI/ISA 95.00.01. Enterprise Control System Integration. Part 1: Models and Terminology.

Bigus Joseph & Jennifer Bigus. Constructing Intelligent Agents Using Java. Second Edition. Wiley Computer Publishing. Canada. 2001.

Bravo, Víctor. Propuesta de un sistema de gestión de servicios para el SCDIA. Universidad de los Andes. Mérida. 2003.

Jennings, N. R., et al. Autonomous Agents for Business Process Management. researchindex.org

Mousalli, Gloria. Modelo de Referencia para el Desarrollo de Sistemas de Control Distribuido Inteligente basado en Agentes (SCDIA). Universidad de los Andes. 2002.

www.fipa.org. FIPA (Foundation for Intelligent Physical Agents).