4th WSEAS Int. Conf. on COMPUTATIONAL INTELLIGENCE, MAN-MACHINE SYSTEMS and CYBERNETICS Miami, Florida, USA, November 17-19, 2005 (pp134-139)

Industrial Automation Architecture: An Intelligent Agent Based Approach*

ADDISON RIOS-BOLIVAR[‡] JOSE AGUILAR[‡,∓]

FRANCISCO HIDROBO[∓,§] LEANDRO LEÓN[‡]

[‡]CEMISID, Facultad de Ingeniería

[∓]Laboratorio de Sistemas Inteligentes

[§]Laboratorio SUMA, Facultad de Ciencias

Universidad de Los Andes

5101. Mérida, VENEZUELA

Abstract: - The purpose of this paper is to present an automation architecture based on intelligent agents. Starting from the functional requirements of a process automation system, a hierarchical automation architecture is formulated in order to down intelligence to inferior levels. This advantage is possible grace to multi-agent systems, which allow the implementation of interoperativity, heterogeneity and complexity due to the requirements of industrial automation. The architecture is composed by a superior layer, and a middleware layer. The superior layer is constituted by two multi-agent systems, which are based on the control and automation functionalities. The middleware layer is designed in order to support superior layer activities and manages the communications from/to field devices. This layer satisfies the FIPA specifications, it also contains two subsystems: one to handle the communications among different sites, and other to give management services of agents. The middleware can support any multiagent system, and has great design versatility.

Key-Words: - Agent applications, Automation systems, Process control software, Plant design, Plant engineering.

1 Introduction

The industrial plants demand every day sophisticated systems that allow them to guarantee a reliable and highly profitable production. Thus, process automation systems are applications that have been characterized by requirements emphasizing safety, reliability, efficiency and quality [1]. Therefore, the automation systems are complex, large, distributed and persistent hardware software systems being stamped by the characteristics of the technical processes they are designed to control [7].

The rapid development of high-capacity hardware components and the information and communication technologies leads to an increasing complexity and a strong need for integration in industrial automation (AI). Traditionally, the automation systems have had a hierarchical structure with different automation functions at each hierarchical level.

*This work was supported in part by Agenda Petróleo funds under grant project 97003817 FONACIT-Venezuela and by the CDCHT-ULA.

This hierarchical structure is called the automation pyramid and represents a canonical abstract model of the industrial complexes [8]. Thus, automation systems should be able to handle complex interactions between hardware and software entities. The software capabilities to manage the evolving complexity are referred to intrinsic properties of all automation systems, which can be summarizes: automation systems are complex and distributed systems; they require different views; the software must be flexible and adaptable.

By the another hand, new paradigms have arisen in the design of computational tools. For example, the agent-oriented paradigm permits to design complex and sophisticated software systems. A software agent is a *proactive object*. The decision about how and when to perform an action is controlled by the agent itself. Beyond this, it is able to execute an action autonomously without being invoked externally. This quality diverges from a passive software entity like a software component which waits for a remote interaction [2]. Their most important proprieties are: autonomy, communication, sociability, reactability, intelligence, and mobility. These characteristics allow that agent technology could be used to fulfill the requirements of process automation systems.

2 Industrial Automation and Agent Technology

Process automation has not typically been an early adopter of new information technologies like software agents. However, some research has emerged concerning the application of agent technology to the implementation of process automation systems [2, 3, 9, 10]. This application has been characterized by the match between the operational principles of process automation systems and agents, where complex and distributed systems of engineering can be obtained.

The automated systems can be represented in different levels, each one has adequate operation characteristics: one level for field devices in order to capture process information, one level for supervisory control and optimization where the tasks of process control are executed, and another level for process management where the production strategies are evaluated and developed. This architecture of hierarchical operation allows the distribution of functionalities of the automation activities through the description of different strategical, tactical, and operational tasks, where the intelligence is centralized in the superior levels.

In that same sense, the agent-oriented approach is a natural way of system decomposition and a reasonable alternative to contemporary approaches in software engineering [9]. The levels of a automatized system can be represented by subsystem components, which are mapped to agents and agent communities (MAS); the interactions between subsystem components are mapped to cooperation, coordination and negotiation mechanisms; this same way the organizational relationships are represented. Thus, intelligence can be distributed through the different levels. See Figure 1. Therefore, the agent-oriented paradigm is suitable to meet the requirements of modern automation systems, where the reconfigurability and flexibility are important aspects [5, 6, 8, 10].

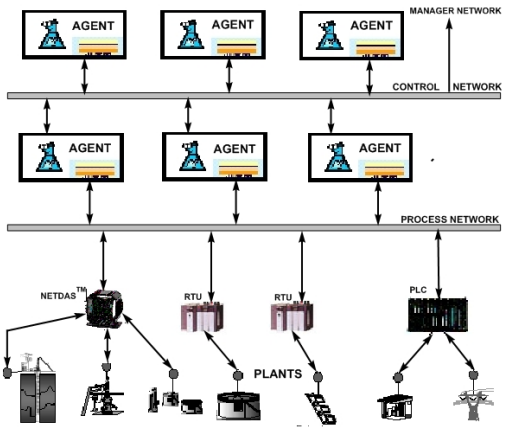

Figure 1: Agents Topology in Automation

3 IA Architecture based on Agents

Just as it is shown in the Figure 1, the control automated systems can be characterized, by their great heterogeneity, by two aspects: the first is the hardware; which is constituted by components of very low level, i.e. remote, PLCs, etc, up to very complex platforms of high level, i.e. industrial computers, servers, clusters, etc. The second level is represented by application software, which is constituted from configurations of field communication protocols, viewers, including pertinent applications to the business. Then, such as the MAS, control automated systems are complexes and heterogenic.

Based on the functional relationship between the industrial automation process and the agent technology, we propose an architecture for automation that consider agents in order to lower and distribute the intelligence to different levels. The proposed hierarchical automation architecture has the following different layers and components (see Figure 2):

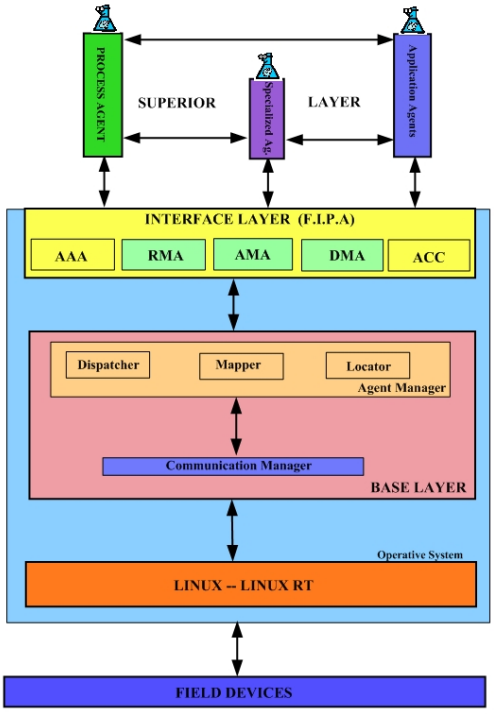

Figure 2: Hierarchic Architecture

- A superior layer. In this layer exists two agents communities: the process and the application agents communities.
- An interface layer. It offers services to the agents communities and its architecture and functionalities are based on FIPA specifications [4].
- The base layer. This layer contains two subsystems: one to handle the communications among different sites, and another one to give agent management services.
- An operating system layer. This layer is represented by the real time operating system installed in the industrial PC. This can be any real time operating system. In our case, an open source operating system based on LINUX is used.

The interface layer and base layer can be treated as an unique layer. This unified layer (middleware) manages resources and services for superior layer. The process agent models the elements of the real world, the application agents execute specific automation tasks, and the middleware agents provide the basic functionalities to manage a Multiagent System.

3.1 Functions in superior layer

Our automation system is a complex information system with automation tasks. In this sense, the automation system have to make, at least, these functions:

1. Monitoring a set of operation variables in a dynamic way. This allows to detect and manage emergencies and production problems in the plant.
2. Management of the complex interactions among the operation variables to transform them into control commands.
3. Transmission of the control commands to a set of actuators located in the plant.

4. Implementation of economic criterias that can be applied as control commands or as a part of the plant programming functions to improve productivity.

5. Reconfiguration of the control and production systems to fulfill the requirements in one specific moment.

6. Giving information to management and operations personnel about the plant situation and the fabricated products.

7. Simplification of the production and operation data, as well as of the quality data of products, with the intention to form historical data bases that could be used like references to make engineering or marketing decisions.

8. Scheduling of the production to reach the user necessities, maintaining the highest productivity to the smallest cost. In addition, the system must allow the planning of the appropriate maintenance functions (preventive and corrective).

9. Determination of the inventory levels and suitable use, as much for the materials, energy, spare parts, goods in process and products to reach wished economic and productions levels.

10. Providing of interfaces with the external actors that interact with the plant production system, such as accounting, trade, investigation, development and engineering, external transport, suppliers and salesmen, purchases, clients and contractors, corporative managements, etc.

These functions can be distributed and expressed through a hierarchic logical structure. Figure 1 shows an *Automation Topology based on Agents* where the functionalities are distributed with certain hierarchies and each agent has an specific role.

In the superior layer there are two agent communities: *Process Agents*, *Application Agents*. The community of process agents, that represents components and abstractions of real processes (pump, oil well, machines, etc.), is composed by agents based on the physical division of the controlled process and the functional division of agent tasks. This functional representation of the processes through agents has the advantage of executing intelligent tasks, for example, evaluation of parameters and physical variables, performance comparison, etc. The process agents are distributed hierarchically so the total plant can be represent for an agent. Several sub-process agents represent, then, sub-systems of the plant. This descompositional arrangement allows to support abstraction of information, allowing to the process agents of superior level (or higher) to exchange refined knowledge of the process. The main function of the process agents is to maintain the knowledge about the state of the area under its responsibility. The agents produce a refined notion of the state, based on the measured variables of their domain.

The community of application agents make specific functions for industrial automation (visualization, supervision, control, optimization or any specialized function). Specialized agents can require of wrappers. Thus, the application agents are conceived as an specialized multi-agent system for coordination, execution, and evaluation of control, supervision, optimization and administration tasks, which is necessary in the prosecution of the process information and the taking of decisions. Then, the application agents offer services to the process agent community. Both communities interact with the middleware through the interface layer.

3.2 Middleware Layer

The middleware is the basic group of software modules that implant the minimum abstractions for the specification, installation and manipulation of agents and objects.

To assure the operability of the automation system, due to the heterogeneity and com-

plexity, the middleware proposal has a multilayer focus, which is composed of homogeneous layers for the operating system, communications management, and agents management. In general, the middleware is formed by three layers: interface layer, base layer and operating system layer. The Interface layer is based on FIPA specification [4]. It is constituted by five agents: Agent for Administration of Agents (AAA), which is called Agent Management System in FIPA, Resource Management Agent (RMA), Application Management Agent (AMA), Data Management Agent (DMA) and Agent for Communication Control (ACC). The RMA, AMA and DMA agents are specializations of the Directory Facilitator (DF) defined by FIPA. This layer manages the operations and communications in the MAS.

The *base layer* contains two subsystems: one to handle the communications among different sites, and other to give services to the agents. It allows agent localization and association of agents with operating system processes. Also, it gives distributed characteristics to the system implementing some properties of these types of systems: migration, interoperativity, naming, etc. The base layer is formed by two modules: the *Agent Manager* and the *Communication Manager*. The *Agent Manager* implements the different services required to create, delete and manage agents, which are viewed as processes in the computational platform. At this level, naming and migrate characteristics of agents is implemented. The *Communication Manager* allows the communication among different processes, in the same site or in different sites. The Communication Manager must provide reliable communication with the network oriented to invocation, and it is here that interoperativity characteristic of agents is implemented.

The Agent Manager is structured in three sub-modules: Dispatcher, Mapper and Locator:

- The dispatcher module allows the invocation among agents.
- The mapper module allows mapping of agents in a process.
- The locator module allows the location of the agents in the system.

Figure 3 shows the components of the middleware layers. In order to accomplish all their activities, the agents of interface layer must invoke these modules.

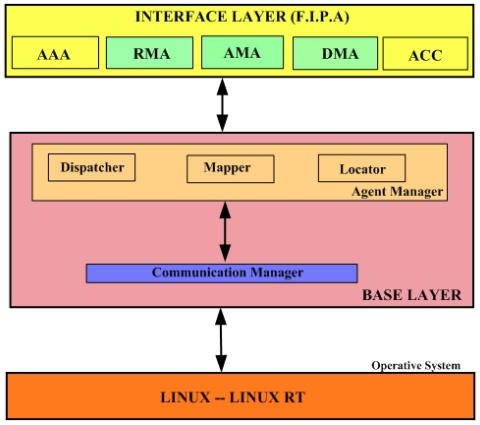

Figure 3: Middleware Implantation Approach

The operating system composes the lowest layer. Normally must be an operating system that support real time. We propose LINUX, which is a popular operating system, that is executed in most of the hardware architectures, and offers support for real time and embedded devices based on firmware. LINUX can give a common layer to all automation hardware.

LINUX, as well as most of the operating systems, base their reliable remote communication on TCP/IP and UDP/IP protocols, that difficult to reach the performance requirements and real time communication constraints. For that reason, we propose additional communication services: rendezvous, asynchronous messages, synchronous messages and RPC messages. In all the cases, the communication is reliable and minimizes the number of network packages transmitted.

4 Conclusion

In this paper we have proposed an automation system based on agents. Our system exploits the main features of the multi-agent systems, like autonomy and intelligence of agents, among others. In this way, we describe the automation process like and intelligent distributed system that support the heterogeneity and interoperativity of its components transparently.

Our system has three components, two agent communities to describe the typical automation processes and tasks, and a middleware to support industrial automation activities. Specifically, our middleware give support to automation systems based on agents following the FIPA standard.

The process agent community describes the components of a process automation system, and the application agent community describes the different automation tasks (for example, optimization, visualization, etc.). This is an innovative approach to represent automation process models with large scalability and adaptability capabilities.

Our middleware is composed of layers, where each layer can be defined individually. The only requirement is the set of services that each layer must provide to others. That is, each layer has a set of functionalities. The lowest function is the kernel of the operating system, the second one is the extension of the operating system to support distributed system, and finally, the last one manages the multi-agent system. This last one has been designed using the FIPA standard. In this way, our middleware can support any multi-agent system, and has a great design versatility.

References:

[1] J. Aguilar, A. Rios, F. Rivas, O. Teran, L. Leon, and N. Perez, *Definición de dominios y paradigmas en una arquitectura de automatización industrial*, Tech. Report 05-04, Fundacite-Mérida, Mérida, 2004.

[2] Martin Albert, Thomas Längle, Heinz Wörn, Michele Capobianco, and Attilio Brighenti, *Multi-agent systems for industrial diagnostics*, XV IFAC World Congress (Barcelona, Spain), 2003.

[3] Alexei Bratoukhine, Yoseba Penya, and Thilo Sauter, *Intelligent software agents in plant automation*, Tech. report, Vienna University Of Technology, Wien Austria, 2002.

[4] Foundation for Intelligent Physical Agents, *www.fipa.org*.

[5] Nicholas Jennings and Stefan Bussmann, *Agent-based control systems*, IEEE Control Systems Magazine **June** (2003), 61–73.

[6] Axel Klostermeyer, *Revolutionising plant automation the pabadis approach - white paper*, Tech. Report IST-1999-60016, PABADIS: Information Society Technology, Germany, 2003.

[7] Jim Pinto, *Instrumentation & control on the frontiers of a new millennium*, Journal Instruments & Control Systems **1** (2000), no. 2, 78–90.

[8] Ilkka Seilonen, Teppo Pirttioja, and Pekka Appelqvist, *Agent technology and process automation*, Tech. report, Helsinki University of Technology, 2003.

[9] Thomas Wagner, *An agent-oriented approach to industrial automation systems*, Tech. report, Institute of Industrial Automation and Software Engineering, University of Stuttgart, 2002.

[10] ———, *Applying agents for engineering of industrial automation systems*, MATES 2003 (M. Schillo *et al., ed.), Springer-Verlag, Berlin Heidelberg, 2003, p. 6273.*