

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



The Role of Business-to-Control Agents in Next Generation Automation Enterprise Systems

Francisco P. Maturana and Eugene Liberman
Advanced Technology Lab
Rockwell Automation
1 Allen Bradley Drive, Mayfield Heights, Ohio 44124

Abstract

Control system level agent technology is a powerful environment that fosters cooperation of control level applications (agents) to solve a set of complex problems not easily solved with a standard control system programming such as ladder code, function blocks, etc. alone. The next logical step in the agent technology evolution is to extend the agent technology capability to the enterprise level. There are many benefits in spanning the agent capability beyond the control system level. The control system environment is not a resource rich environment and some complex large scale applications that require a great deal of computing power may not fit in it. The enterprise level agents can take advantage of the virtually unlimited resources at the enterprise level. This paper discusses the characteristic of intelligent technology and how the agents communicate and cooperate in the different levels of the automation systems.

1. Introduction

Systems in general are becoming more complex and intertwined (Shen et al., 2001) (Mařík et al., 2001). Classical programming and data management will not be able to cope with increased level of complexity. Computing platforms are required to operate at a faster speed to carry out more and more transactions per second. Information integration plays a fundamental role to achieve a more efficient data retrieval and management system. Discovery of information and establishing dependencies among the components of the system is a cumbersome task.

As we progress into the hyper information space realm, automation systems will become more and more involved in the information processing cycles. Automation processes are going to be another node in a global information infrastructure network of resources and, therefore, will need to be more intelligent, highly adaptable, discoverable and information friendly systems.

We envision that in order to fulfill a seamless integration of the distinct levels of the enterprise, we will need to move away from the typical centralized server approach into a

more fine-granular domains with software components acting as independent intelligent agents (Wooldridge and Jennings, 1995)(Brooks, 1986)(Shen *et al* 2001)(Christensen, 1994)(Mařík *et al.*, 2001).

Control system level agent technology is a powerful environment that fosters cooperation of control level applications (agents) to solve a set of complex problems not easily solved with a standard control system programming such as ladder code, function blocks, etc. alone. Rockwell Automation had successfully demonstrated the power and ease of solving complex problems in industrial automation environment using control level agent technology (Maturana *et al.*, 2008)(Gianetti *et al.*, 2006)(Discenzo *et al.*, 2001)(Staron *et al.*, 2004)(Tichý *et al.*, 2002)(Maturana *et al.*, 2004).

The next logical step in the agent technology evolution is to extend the agent technology capability to the enterprise level. There are many benefits in spanning the agent capability beyond the control system level. The control system environment is not a resource rich environment and some complex large scale applications that require a great deal of computing power may not fit in it. The enterprise level agents can take advantage of the virtually unlimited resources at the enterprise level.

Control systems are generally designed for the “factory floor” environment and integration of the “factory floor” functionality with the rest of the computing environment had always been a challenging task. Therefore, the creation and deployment of the Enterprise level agents that become full members of the control level agent community is a valuable addition to the agent-based control system technology.

Given this additional autonomy at the equipment level, the physical system effectively becomes more survivable and requires less maintenance. The equipment can operate autonomously independent of the rest of the system if a critical event interrupts communications. This property of continuing operation without central control is a must in industrial and military environments.

For example, the Office of Naval Research (ONR) and the US Navy were looking for a highly survivable robust control environment for the chilled water distribution application, a critical ship system (Maturana *et al.*, 2000). One of the major requirements was that the chilled water system continues to operate even after a major disturbance such as an explosion or a missile strike somewhere on the ship occurs. Several approaches were investigated and a distributed intelligent multi-agent system was selected. The main goal was to have a fully distributed system with no single point of failure.

Agents were deployed in the automation controllers. These agents consisted of both reasoning and real-time control and were distributed among 23 controllers which were physically located near the controlled hardware. The reasoning part of the agents inside the controllers negotiated the control actions to accomplish specific missions and configurations.

Figure 1 shows the Navy’s land-based water cooling system testbed that we prepared with controller enclosures to download the agents. The testbed included real plumbing, controls and communications, and electrical components that resembled the real ship systems but in a reduced scale. A typical control plan consisted of water routes to transport cold water from the cooling units into the heat loads (computers, radars, weaponry, etc.) and water routes to move hot water from the loads back to the coolers.

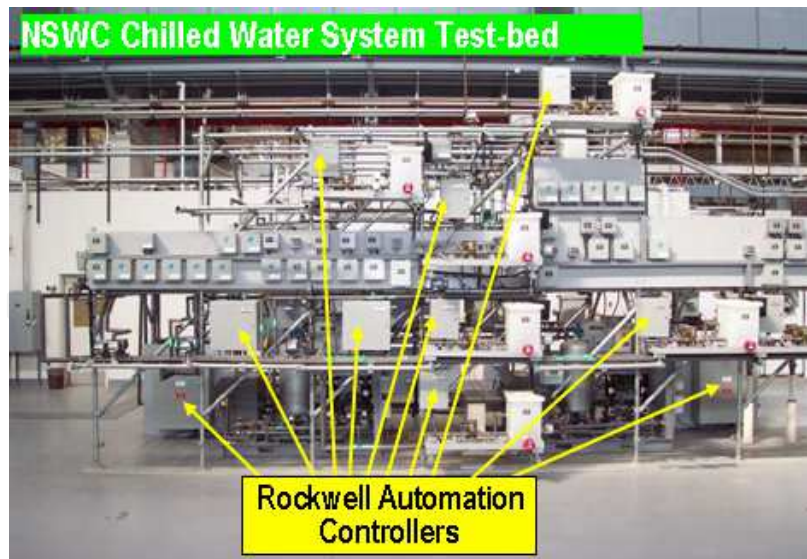


Fig. 1. Office of Naval Research chilled water land-based simulator

The agents evaluated a number of conditions that affected the physical device in order to create a feasible water route. Sometimes lines were obstructed to simulate missing capability, but the agents evaluate adapted their decisions to discover feasible alternatives without following prescribed configurations or pre built tables.

The land based simulator helped in understanding how to program intelligent software in embedded control devices. The agent offered a new dimension in adaptable control systems. However, this capability was limited to device level only. To date we confront a different set of requirements that involve a multi tier system architecture in which we are being asked to combine requirements and capabilities from different layer of the enterprise. Our intention is to explore and demonstrate the architecture of the business-to-control layers and how these will be constructed and interfaced to the different tiers of the manufacturing and information systems, the industrial environment.

2. Intelligent agent architecture

An important objective of the distributed artificial intelligence and cognition research is to provide a foundation to enhance the capabilities of machines and to make machines more useful and intelligent (Nilsson, 1980)(Balasubramanian et al., 2000)(Brennan et al., 2003)(Charniak and Mc Dermott, 1985). Some of the techniques pursued include a suite of Artificial Intelligence (AI) techniques such as expert systems, fuzzy logic, genetic algorithms, reasoning, artificial neural networks, and model-based techniques. Many of the automation successes reported applied biologically inspired architectures (Brooks, 1986)(Christensen, 1994) and techniques to solve well-targeted automation problems such as adaptive control, defect classification, and job scheduling to name a few.

The capabilities which may be provided by intelligent machines may be categorized based on the degree of embedded knowledge with the most capable systems employing real-time goal adjustment, cooperation, pre-emption, and dynamic re-configuration. These capabilities may be effectively integrated in an agent-based system employing intelligent machines in a distributed automation system. This architecture is built on a foundation of a

society of locally intelligent cooperating machines. It provides an effective framework for an efficient and very robust automation of complex systems.

An agent-based control solution is built around a set of application rules and behaviors. As shown in Figure 2, an agent solution has a tree-like hierarchal shape in which each branch and sub branch can be made of agent components and attributes. The structure of an agent is made from three main components: (1) Reasoning, (2) Control, and (3) Data Table.

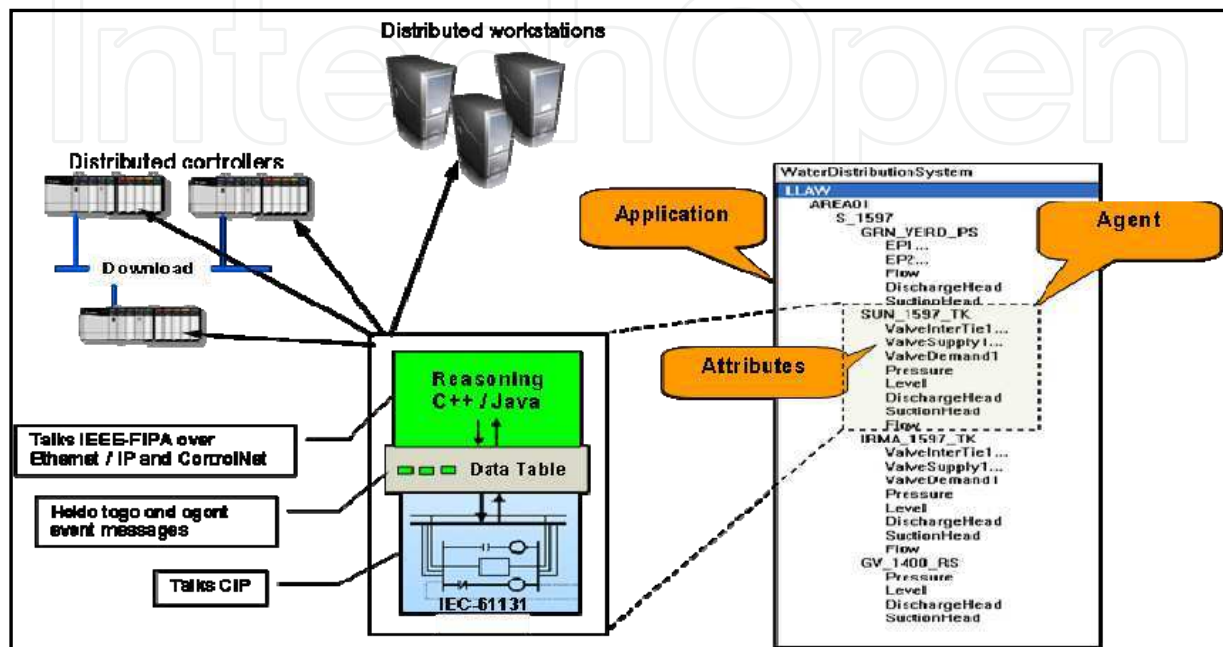


Fig. 2. Agent architecture

The reasoning part is a software component that conveys the agent's heuristics. This component defines the behavior of the agent according to the evolution of its internal rules and the interaction with the control level component. There are event-based transactions between the reasoning and the control level components that are defined during the agent programming phase.

The agent initiates reasoning about a particular event based on the arrival of a global message. A global message is an inter-agent communication that conveys requests or just information. The global message is associated with a particular capability of the agent that is specified as part of the agent behavior. Upon arrival of the global message, the agent behavior for an associated capability is fetched for execution. At this point, the options are diverse since the agent behavior can contain multiple steps that require internal actions as well as the initiation of more global messaging that the agent needs to complete its local goals. A goal is a plan that an agent constructs by pulling together local and combined capabilities. The combined capabilities are obtained via negotiation with other agents. Global messages are encoded according to the Foundation for Intelligent Physical Agents (FIPA, 2000) protocols.

Another way to drive the agent behavior is via the planner engine (see Figure 3). The role of the planner is to coordinate the events from the control level with the agent behavior. Again, there are multiple options on how to do this since the control level events can be associated

with a variety of steps in the reasoning layer. The extent of these associations is a system designer decision.

We use a distributed control architecture based on automation control devices with extended firmware. The extended firmware allows for the realization of component-level intelligence which converts the device into an intelligent node with negotiation capabilities. The intelligence of the application can now be distributed among multiple controllers as opposed to the traditional control system programming in which the concentration of functionality is more predominant.

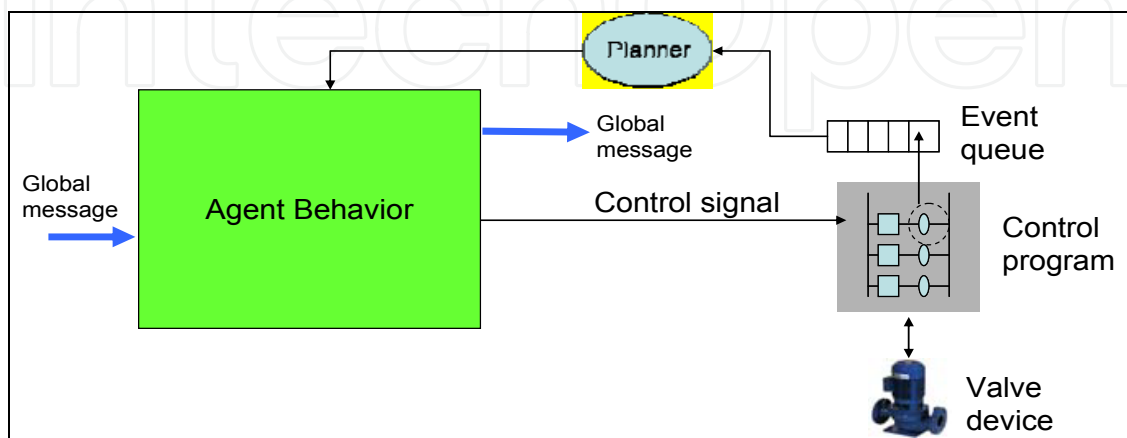


Fig. 3. Agent behavior fetching via the planner

Agents are goal-oriented entities that act autonomously and cooperatively when involved in a problem-solving task. Although an agent is an individualistic entity when pursuing local goals. They organize their individual capabilities around system goals. An agent capability is a description of the type of operations an agent can do. For example, a welding robot agent has a capability to weld specific spots on a structure. On the other hand, a system goal is abstract because there is no explicit declaration of it during the design of the system. A goal can be described as a dynamic social force that pulls agents together to solve a particular task. Thus, a system goal has the following social attributes: (a) System event or need, (b) A first and second level responders, (c) Plan of actions, and (d) Execution.

3. Organizing distributed agents

One of the key benefits of using agents is their ability to work in a distributed environment. Agents use social skills to overcome challenges of the distributed environment.

3.1 Agent design

The effort to program agents may become a difficult task if there are no well defined boundaries between the functions. However, it is always possible to force an initial partitioning to build a working model. The rule of thumb is to generate agents that encapsulate a physical device such as a valve or water pump; these are well defined devices with clear boundaries. But, as the implementation moves into the reasoning layer, it becomes even more difficult to define the boundaries. For example, the designer has to be prepared to decide the pump agent behavior and the context in which pumps negotiate and what they negotiate for.

Later we will discuss the agent wrapper layer. In such a model, control related functions are kept in the control level and encapsulated with a rather simplistic agent. The higher level behaviors can then be modeled in the upper level as business processes that interact with its control counterpart. From a design point of view the latter approach is very appealing since business level processes will count with greater computing resources than the ones provided by the embedded devices. This brings more freedom into the programming of the functions and helps in deciding on performance issues relative to the agent partitioning.

From the agent technology point of view, industrial applications can be designed according to their decision making complexity and size. The decision making complexity of a complex machine has greater magnitude than a simple machine with a reduced number of actuation points and inputs and outputs connections. The other dimension relates to the size of the application which depends on the number of nodes that are needed to model to operations of the manufacturing plant. Thus, it is possible to categorize the control applications according to two these axes, i.e., complexity and size. For example, a material handling system such as a distribution hub is made of a large number of conveyor belts. Each conveyor belt is a simple machine but the material handling operation requires a combination of multiple of these simple machines to carry out the transportation of the material. On the opposite side, we find systems with a reduced number of machines but the machines are very sophisticated in terms of configuration setup, inputs, and outputs. Anywhere in the middle we find a variety of applications that fluctuate between the two poles, as shown in Figure 4.

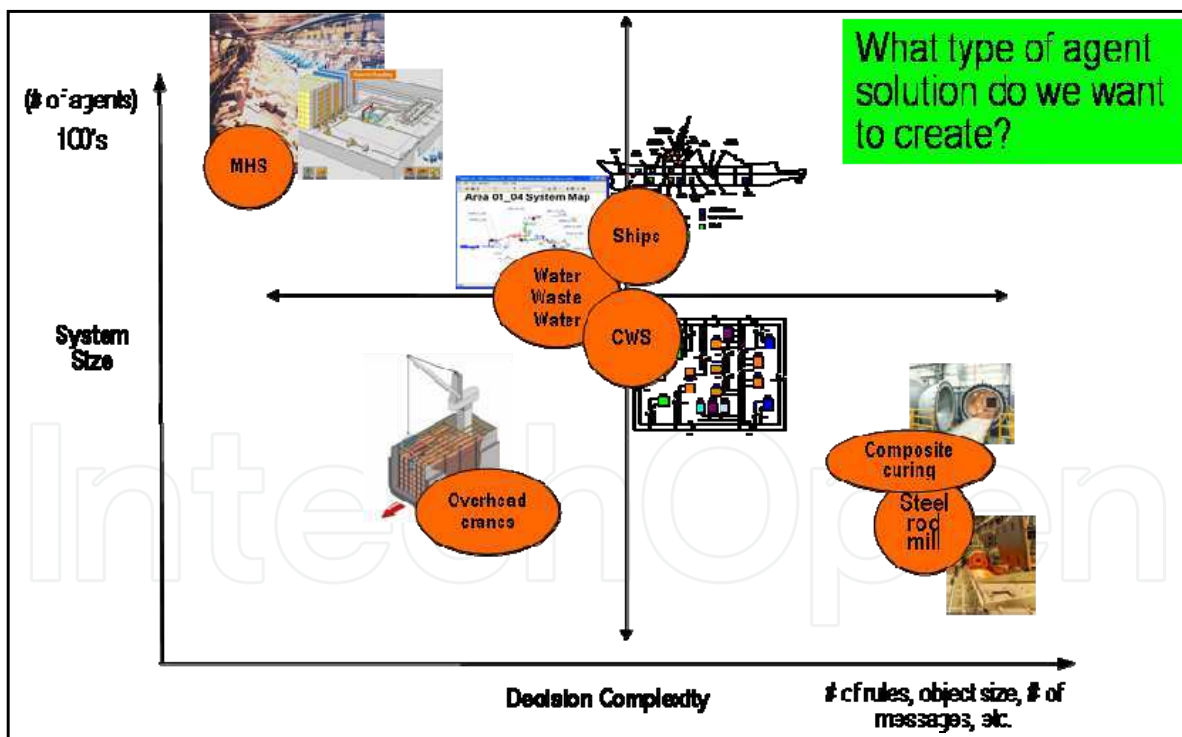


Fig. 4. Application domain categories

Application domain categories are very important in modelling distributed agent control since they frame a better division of functionality. There is a need for establishing rules to design this type of systems. One goal is to highlight one of the most difficult aspects of agent modelling which is the definition of the agent boundaries. Although it is possible to

create centralized, monolithic agents to handle all aspects of the manufacturing organization, it is not a recommended option. We will show that in order to bring greater flexibility and effective scalability into the enterprise, it is required to have a separation of the functions into different layers to make the system more distributed.

4. Enterprise level agent technology architecture

In the interest of simplicity, we will focus the description of our business-to-control architecture as a two layer interaction. In this description, there is an enterprise level and a control level or enterprise domain and control domain.

One of the properties of the control level agents is the capability to find all the peers and establish communication between them. To accomplish agent discovery social knowledge must be composed and stored in directory services. The directory services organize social knowledge using one of the techniques above to propagate agent information throughout the different layers. The social knowledge information can be propagated in an automated fashion or on demand. But, these directory actions take place naturally as part of the directory service functions.

The Enterprise level agents inherit the directory service information from the control level agents. All agents that register their social information with the control level directory services are also known in the enterprise level via social knowledge propagation policies, as shown in the Figure 5. These directory services contain agent properties such as: capabilities, functions, and input and output parameters, etc. This information is required for an application at the Enterprise level to utilize the agent's services in the control level. The LDAP server is fed with this agent information from the Enterprise level DS via an LDAP proxy (LDAP Enterprise agent) at the initialization phase.

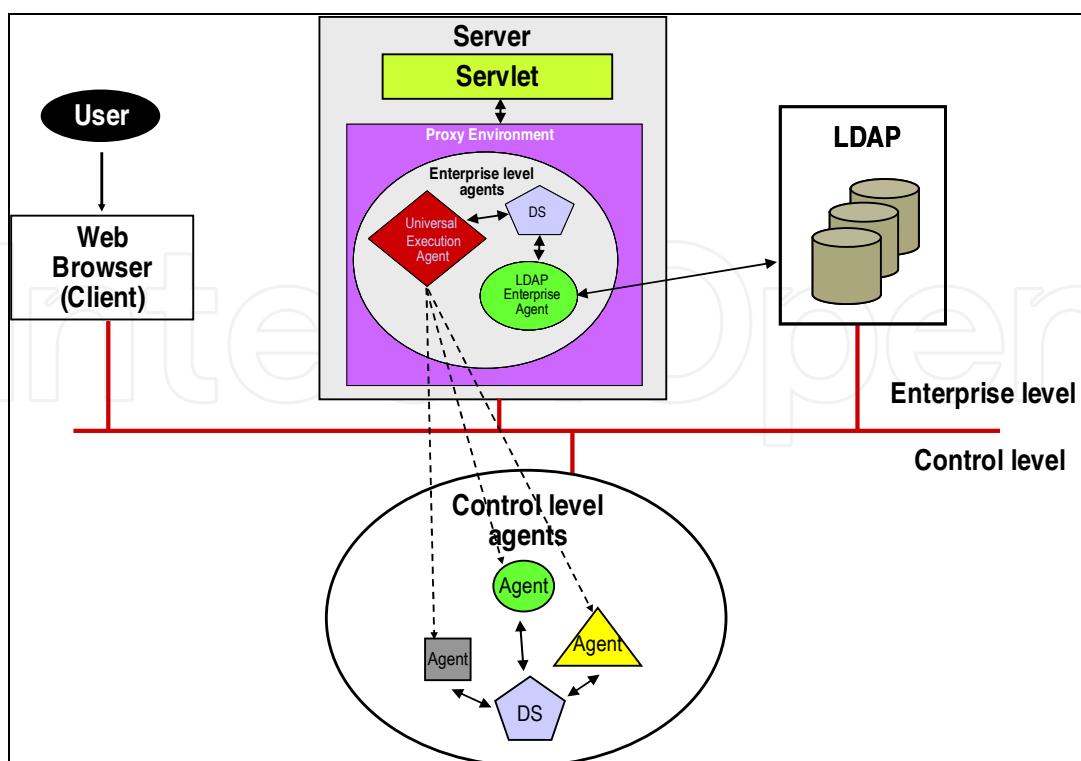


Fig. 5. Directory Service layers

The LDAP directory server was chosen due to its flexibility and accessibility. Any application on the network can access LDAP server, provided that the user or application, an LDAP client, has the proper credentials. LDAP is a standard protocol so every LDAP client adheres to the same standard that is portable and relatively easy to implement.

In this work, we moved the system integration in the direction of a universal model. We are interested in identifying the software components and terminology for the interfaces and communication between the enterprise level and the manufacturing floor. In our business-to-control architecture, we show an initial set of mechanisms that help in connecting the processes without having to be too specific about the information exchange details.

To this end, the agent infrastructure accommodates a proxy environment component that can be dynamically created to bridge the two layers. The proxy environment is a wrapper that knows how to move information across the layers, thereby supporting translation and interpretation of the information.

5. Benefits of the enterprise component to the agent infrastructure

Section 3 and 4 describe the technical requirements and implementation of the enterprise level agents and how the enterprise and control levels can be integrated. This section will show some practical applications of this integrated framework and specific examples will be provided to illustrate the benefits of the interacting layers. We will introduce a water distribution system as an example describing integration of control and enterprise level agents.

In the water distribution system, the control level agents can solve the problem without the “help” of enterprise level agents. But the introduction of the enterprise level agents can increase efficiency and reduce cost of the control level agent system alone. Since control-level agents have limited information about the system, their decision making scope lacks the desired level of optimality. Then, to compensate for the lack of knowledge, the control level agents have been programmed with cooperation protocols to allow them to explore the universe of discourse. Although the cooperative search for solutions may put the agents closer to a near optimum equilibrium, there is still the problem of partial knowledge and localized observation. Thus, the use of enterprise level agents is justified from the point of view of augmented system-level knowledge. Since an enterprise level agent has access to unlimited resources and, services, and databases, it is a much better location to program more exhaustive search nets to support a more global decision making process which can then be coordinated with local level agents.

Furthermore, the enterprise level agents can be launched to report the status of any set of control level agents and all the enterprise level agents can be engaged and controlled from a web browser, so the system status can be remotely obtained at any time from anywhere.

5.1 The BPEL orchestration role

Another benefit of enterprise level agents is the fact that they can be wrapped in web services or other enterprise applications. These web applications can be orchestrated into more complex functions that can run in the background. The orchestrated services then become business level processes that can be coordinated and deployed by a Business Process Execution Language (BPEL) engine (BPEL, 2002). BPEL offers a rich set of features to coordinate business process into an integrated process that oversees the combined

activity for all involve processes, as shown in Figure 7. Concurrency is a natural aspect of the BPEL orchestration permitting processes to execute and communicate in parallel. In our architecture, we can bring the BPEL activity into the agent world as another agent capability since we are able to encapsulate the BPEL process as another agent behavior. BPEL orchestration engines can then be brought into the decision making loops and knowledge exploration in parallel to support the high and low-level agents.

One of these orchestrated applications can be system status monitoring and fault notification. The process status and fault identification can be displayed in a web browser as well. This reduces the requirements on the number of personnel assigned to the role of monitoring the system's status. Monitoring can take place anywhere with Internet connectivity and this has a great potential to reduce costly infrastructure and software development.

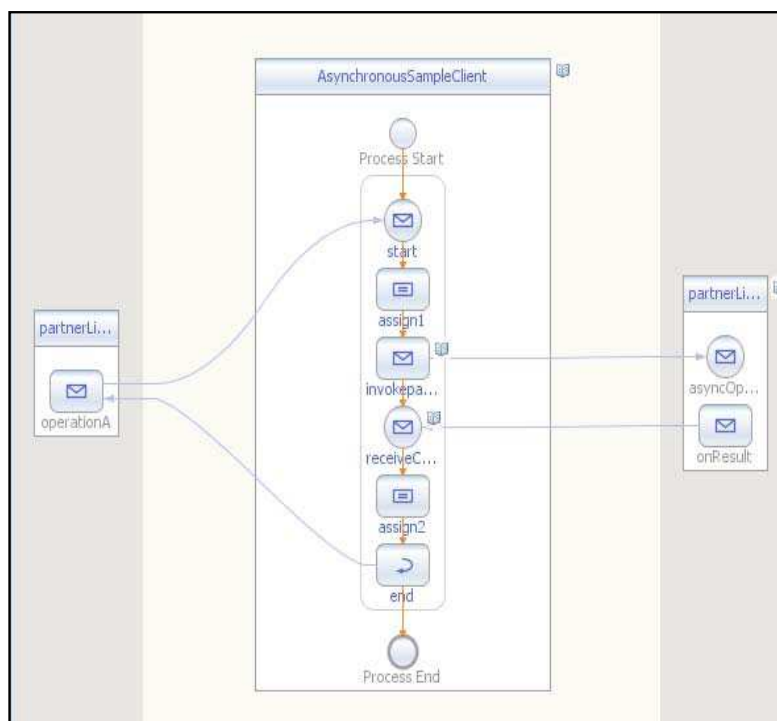


Fig. 7. BPEL orchestration example

The Enterprise level environment has virtually unlimited resources. Several instances of vital redundant agents can be launched and deployed at the enterprise level on various machines. A failure of a machine hosting an agent environment will not impact the system integrity as a whole. The agent system will automatically reconfigure itself and utilize services of available agents. The BPEL orchestration task can be programmed to carry out launching a new instance of an agent if the original instance does not respond or reports failure. The options are unlimited.

6. Sample application: Water distribution and electrical power negotiation

The ability to interconnect the control and the enterprise levels via the business-to-control infrastructure will allow for a consideration of a new breed of control scenarios. We looked

over different aspects of the water distribution domain in which we could demonstrate enterprise and control level agents (Giannetti et al., 2005). We found that in the domestic water distribution systems there is a mix of requirements and decision-making scenarios that map well to the two-level interoperability.

In a domestic water distribution system, there are quality and process requirements that cannot be completely contained in the automation controllers (aka, Programmable Logic Controller – PLC) (CIP, 2001)(IEC, 2001). For example, process-level requirements include water availability, chlorination ratios, residual ratios, etc. Higher level requirements include scheduling of water pumping to accommodate low-cost electricity pricing intervals or seasonal conditions, etc. These requirements shape the design of the agent system in terms of system partitioning and distribution of capabilities. Figure 8 shows the type of agents (green bubbles) that are used in a water distribution system model: pump station and pump, tank, utility company, city water. Each of these agents has the knowledge about how to operate a specific piece of equipment but they also depend on business level knowledge to make high value decisions. In our proposed solution, we include enterprise level services to contain the business level knowledge: utility company and city water. These services provide access to system's historical data (demand, consumption patterns, and electricity pricing policies).

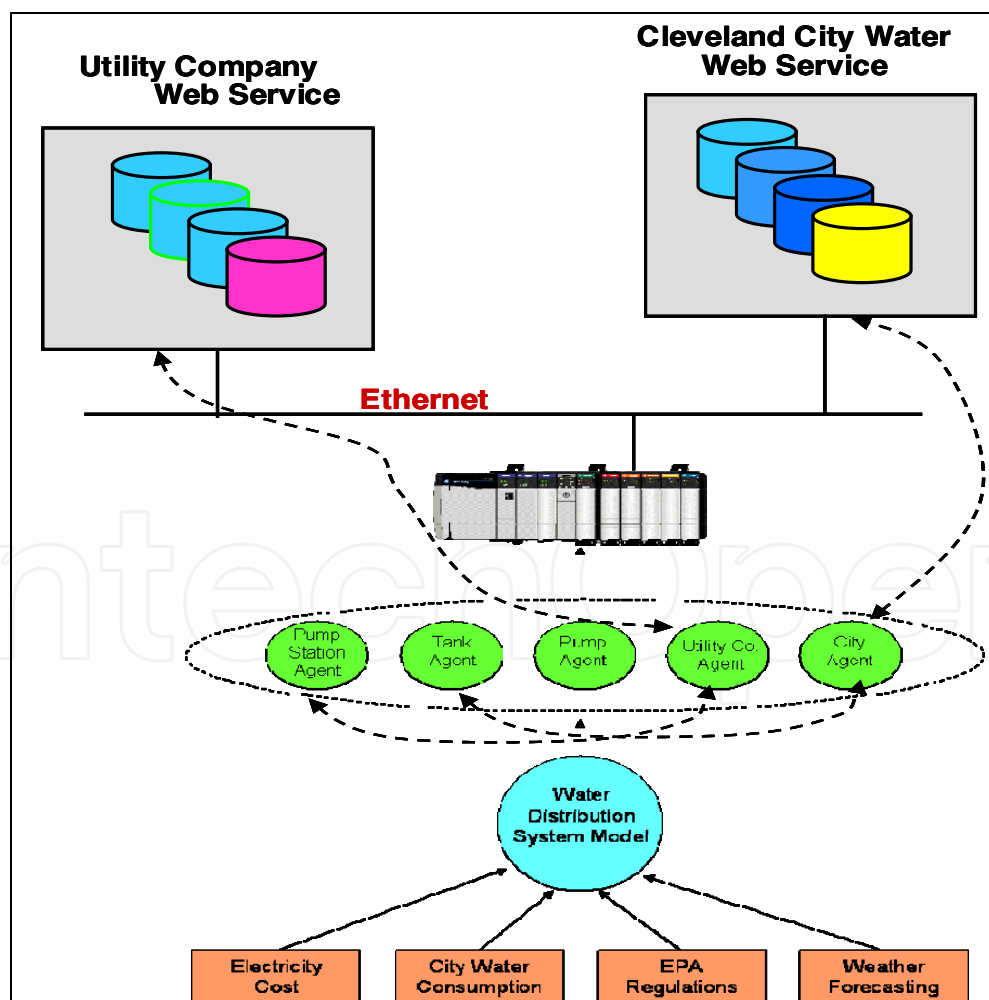


Fig. 8. Agent-based water system

The combined actions of the two levels allows for a complex decision making system. For example, a water tank agent will see the need for receiving additional water (n -gallons) to fulfill some near future demand (i.e., forecast demand). The water tank agent in the PLC needed to converse with the city water service to forecast the water demand for a city district in the near future. The city water service has the necessary computing capacity and access to databases to estimate the demand based on the actual, historical, and seasonal consumptions. Here is where the business-to-control engagement adds its value.

Another scenario also takes place between the water system and the utility company services, as shown in Figure 9. After the water tank agents calculate their future demand, they emit a request for pumping water to the pumping station agents. The pumping station agents need to carry out process level calculations to estimate the amount of power that is going to be needed to provide the water. This process-level evaluation takes place in between the pumping station and the pumps since this information gathering requires health assessment information that is known by the pumping devices themselves.

The pump stations then contacts the utility company services with a request for low cost electricity. The utility company services have capacity to do the calculations by contacting the pricing interval calculators and databases and perhaps humans to estimate a final price for the electricity and a valid time interval for the offering. The utility company service needs to interact with seasonal and historical databases to estimate the prices. This example describes a complex interaction among services and agents. In a classical framework without the considerations that have showed in this article, programming the interactions would be expensive and cumbersome.

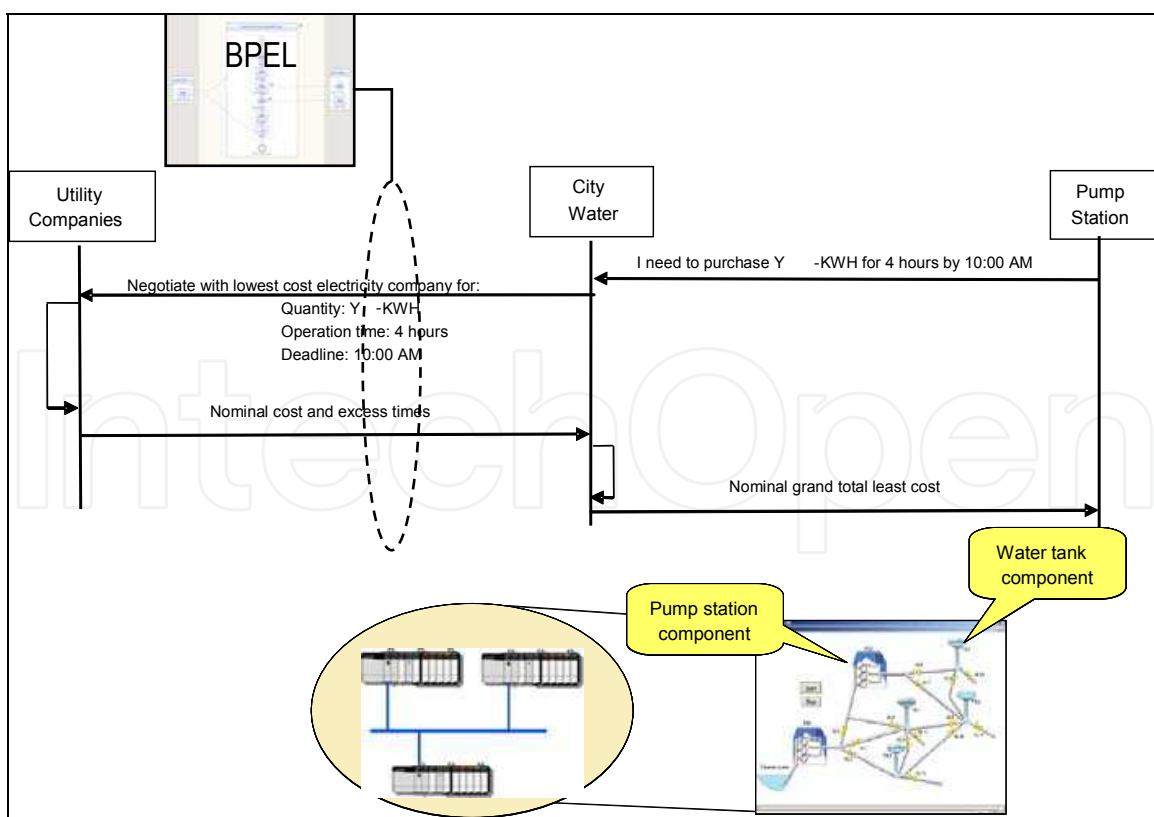


Fig. 9. Complex electricity pricing negotiation

The description above illustrates two representative scenarios that may occur in an agent-based water distribution system between the enterprise and the control levels. The role of the BPEL orchestration is very fundamental in the coordination of the different services and the selection of their responses. There will be multiple transactions going back and forth in multiple directions that need to be coordinated and synchronized in order to maintain the stability of the system. For example, the BPEL process that orchestrates the interaction between the city water agent and the utility companies needs to handle one-to-many transactions in one direction (city water to utility companies) and many-to-one transactions in the opposite direction. In the transition between communications, BPEL needs to listen for the responses while applying system-level rules to decide on the most suitable responses to be emitted to the agents. The scenarios can become more sophisticated and complicated. But the intention of this work is to show how to assemble the architecture for realizing the future vision of autonomous control systems.

7. Conclusions

The integration of the enterprise level agents and control level agents will make systems more robust and operate at lower cost. However, the right balance needs to be maintained between the control and the enterprise functionalities. Systems designers will have to make sure the loss of enterprise capability will not compromise the fundamental control level ability to carry out control tasks autonomously.

8. References

- M. Wooldridge and N. Jennings, "Intelligent Agents: Theory and Practice", Knowledge Eng. Rev., vol. 10, no. 2, 1995, pp. 115-152.
- A. Brooks, "A Robust Layered Control System for a Mobile Robot", IEEE J. Robotics and Automation, vol. 2, no. 1, 1986, pp. 14-23.
- Shen W., Norrie D., and Barthès J.P.: "Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing". Taylor & Francis, London, 2001.
- J. H. Christensen, "Holonc Manufacturing Systems: Initial architecture and standards direction," in *Proc. First European Conference on Holonic Manufacturing Systems*, Hanover, Germany, 1994, pp. 20.
- Mařík, V., Pěchouček, M., and Štěpánková, O., Social Knowledge in Multi-Agent Systems. *Multi-Agent Systems and Applications*, LNAI 2086, Springer, Berlin, 2001, 211-245.
- Francisco P. Maturana, Dan L. Carnahan, Donald D. Theroux, Kenwood H. Hall: Distributed multi sensor agent for composite curing control. ETFA 2008: 1236-1243.
- Lucilla Giannetti, Francisco P. Maturana, Frederick M. Discenzo: Agent-Based Control of a Municipal Water System. CEEMAS 2005: 500-510.
- Discenzo, F.M., Marik, V., Maturana, F.P., Loparo, K.A., 2001, *Intelligent Devices Enable Enhanced Modeling and Control of Complex Real-Time Systems*, International Conference on Complex Systems, (Irvine).
- Staron, R. J., Maturana, F. P., Tichý, P., and Šlechta, P., Use of an Agent Type Library for the Design and Implementation of Highly Flexible Control Systems. *The 8th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2004)*, Orlando, USA, 2004, 18-21.

- P. Tichý, P. Šlechta, F. P. Maturana, and S. Balasubramanian, "Industrial MAS for Planning and Control," in (Mařík V., Štěpánková O., Krautwurmová H., Luck M., eds.) *Proc Multi-Agent Systems and Applications II: 9th ECCAI-ACAI/EASSS 2001, AEMAS 2001, HoloMAS 2001*, LNAI 2322, Springer-Verlag, Berlin, 2002, pp. 280-295.
- F. Maturana, P. Tichý, P. Šlechta, F. Discenzo, R. Staron, and K. Hall, "Distributed multi-agent architecture for automation systems", *Expert Systems with Applications* 26, 2004, pp. 49-56.
- Maturana F., Balasubramanian S., and Vasko D.: *An Autonomous Cooperative Systems for Material handling Applications*. ECAI 2000, Berlin, Germany, 2000.
- Francisco P. Maturana, Raymond J. Staron, Kenwood Hall: *Real time collaborative intelligent solutions*. SMC (2) 2004: 1895-1902.
- Charniak, E., & McDermott, D, 1985, *Introduction to Artificial Intelligence*, Addison-Wesley.
- Nilsson, N., 1980, *Principles of Artificial Intelligence*, Morgan Kaufmann (Los Altos).
- S. Balasubramanian, R. W. Brennan, D. H. Norrie, "Requirements for Holonic Manufacturing Systems Control," in *Proc DEXA Workshop 2000*, 2000, pp. 214-218.
- R. W. Brennan, K. H. Hall, V. Mařík, F. P. Maturana, and D.H. Norrie, "A real-time interface for holonic control devices," in: V. Mařík, D. McFarlane, P. Valckenaers (Eds.): *Holonic and Multi-agent Systems for Manufacturing, Lecture Notes in Artificial Intelligence*, No. 2744, Springer-Verlag, Berlin, 2003, pp. 25-34.
- FIPA, The Foundation for Intelligent Physical Agents (FIPA), 2000: <http://www.fipa.org>.
- CIP: Common Industrial Protocol, Available, 2001: http://www.ab.com/networks/cip_pop.html.
- IEC (International Electrotechnical Commission), TC65/WG6, 61131-3, 2nd Ed., *Programmable Controllers - Programming Languages*, April 16, 2001.
- Business Process Execution Language for Web Services Version 1.1., July 2002, <http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel/ws-bpel.pdf>.

IntechOpen

IntechOpen

IntechOpen



Factory Automation

Edited by Javier Silvestre-Blanes

ISBN 978-953-307-024-7

Hard cover, 602 pages

Publisher InTech

Published online 01, March, 2010

Published in print edition March, 2010

Factory automation has evolved significantly in the last few decades, and is today a complex, interdisciplinary, scientific area. In this book a selection of papers on topics related to factory automation is presented, covering a broad spectrum, so that the reader may become familiar with the various fields, and also study them in more depth where required. Within various chapters in this book, special attention is given to distributed applications and their use of networks, since it is one of the most relevant subjects in the evolution of factory automation. Different Medium Access Control and networks are analyzed, while Ethernet and Wireless networks are looked at in more detail, since they are among the hottest topics in recent research. Another important subject is everything concerning the increase in the complexity of factory automation, and the need for flexibility and interoperability. Finally the use of multi-agent systems, advanced control, formal methods, or the application in this field of RFID, are additional examples of the ideas and disciplines that experts around the world have analyzed in their work.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Francisco P. Maturana and Eugene Liberman (2010). The Role of Business-to-Control Agents in Next Generation Automation Enterprise Systems, *Factory Automation*, Javier Silvestre-Blanes (Ed.), ISBN: 978-953-307-024-7, InTech, Available from: <http://www.intechopen.com/books/factory-automation/the-role-of-business-to-control-agents-in-next-generation-automation-enterprise-systems>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen