

HOLONIC AND AGENT-BASED CONTROL

Kenwood H Hall, Raymond J Staron, Pavel Vrba

Rockwell Automation, Inc.

Abstract: Currently industrial automation systems are built using a hierarchical top-down approach, yielding tightly coupled and low flexibility systems. Holonic and intelligent agent-based industrial control systems have the potential to be much more highly robust and flexible systems with very loose coupling between subsystems. This paper explores Rockwell Automation's current agent philosophy, application experience, and the obstacles to widespread adoption of agent technology in industrial automation systems.
Copyright © 2005 IFAC

Keywords: Agents, Automation, Autonomous Control, Configuration Control, Decentralized Control, Distributed Control, Flexible Automation, Intelligent Control, Methodology, Programmable Controllers, Robust Control, Systems Methodology.

1. AGENT PHILOSOPHY

Holonic manufacturing systems (HMS) represent a novel paradigm to address some critical problems faced by manufacturing businesses in the twenty-first century. Ever increasing customer requirements are calling for new manufacturing strategies satisfying the needs for (i) open and dynamic structures to allow the on-line integration of new subsystems or removal of existing subsystems from the system without the need to stop and reinitialize the working environment, (ii) agility to adapt quickly to continuous and unanticipated changes in the manufacturing environment, and (iii) fault tolerance to detect and recover from a failure by minimizing its impact on the whole system.

Distributed intelligent manufacturing can meet these requirements. The more traditional sequential and centralized solutions, used within the scope of such agile environments, do not work since they are slow to react, impose operational bottlenecks, and are a critical point of failure. Holonics is a decentralized 'bottom up' approach and provides principles to ensure a higher degree of responsiveness and handling of system complexity. The fundamental building blocks of a HMS are called *holons*, fundamentally as presented by Koestler (1967), to reflect the fact that these entities: (i) are both parts

and wholes and (ii) behave simultaneously in an autonomous and cooperative fashion.

The vision of a *holonic factory*, in which all the operations (including product ordering, planning, scheduling, manufacturing, and invoicing the customer) are based entirely on holonic principles, covers several levels of information processing for manufacturing. At least three separate levels can be distinguished:

- **real-time control**, tightly connected with the physical level of manufacturing equipment;
- **production planning and scheduling** both on the workshop and factory level;
- **supply chain management**, integrating the particular plant with its external entities (suppliers, customers, partners, sales network etc.).

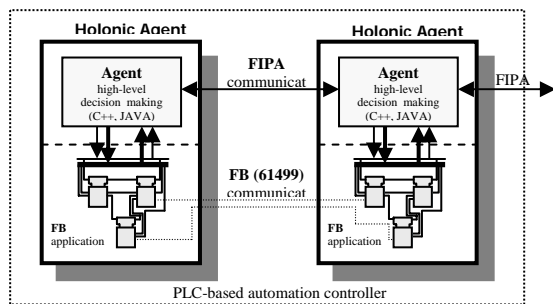


Fig. 1. Model for holonic agents using 61499 function blocks for real-time control.

The particular research results in the holonic field are connected mainly with real-time control. In the other two subfields, the research centers on the philosophical or architectural level, but the particular implementations exclusively use the research results of multiagent systems (MAS). The HMS community has fully realized that the *function block* based real-time control (utilizing the IEC 61499 standard) is applicable to only very limited control tasks and that it is necessary to leverage the results achieved in the MAS field to widely exploit the visions of a holonic factory. Several general architectures for combining both the function block and MAS technologies have been designed. The most popular holon model encapsulates one or more function block oriented devices into a wrapper containing a higher-level software component (see Figure 1).

Rockwell Automation (RA) has realized that the 61499 function blocks are not as ubiquitous as the IEC 61131-3 programmable controller languages described in (IEC, 2001), and therefore has implemented its multiagent system using the most popular IEC 61131-3 language in North America, relay ladder logic. The extension of these concepts to the other IEC 61131-3 languages appears feasible but was outside the scope of this project. The RA model for its *holonic agents* (or simply, *agents*) is still one containing a higher-level intelligent software component, as shown in Figure 2.

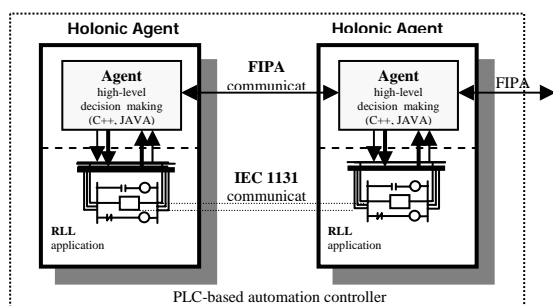


Fig. 2. Model for RA holonic agents using IEC 61131-3 relay ladder logic for real-time control.

In such an agent, equipped with both a lower level real-time component and a higher-level intelligent component, there are three communication channels:

- intra-agent communication between the real-time component and the intelligent component; RA has implemented its agents on its Logix brand of controllers, and the data tables are used for communication between the two components;

- inter-agent communication that is aimed at communication among the intelligent components of multiple agents; RA uses the Foundation for Intelligent Physical Agents (FIPA) standards with its own Job Description Language (JDL) as the content language;
- a direct communication channel, among real-time components of the neighboring agents was found necessary to meet the real-time constraints of the applications, although it does increase coupling between holons; RA uses the CIP standard for these high speed, deterministic communications.

These agents can therefore widely communicate among themselves, carry out complex negotiations, cooperate, develop manufacturing scenarios, etc., as well as control the manufacturing equipment.

Selling the use of agent-based control systems to customers has proven somewhat problematic. In general, a MAS offers several general benefits relative to a conventional control system:

- **Autonomy:** Each intelligent agent makes its own decisions and is responsible for carrying out its decisions toward successful completion.
- **Cooperation:** Intelligent agents combine their capabilities into collaboration groups to adapt and respond to diverse events and mission goals.
- **Communication:** Intelligent agents share a common language to enable the cooperation.
- **Fault tolerance:** Intelligent agents possess the capability to detect equipment failures and to isolate failures from propagating.

In addition to the general benefits of MAS, the RA implementation has some unique additional benefits:

- The agents are implemented in the controller, eliminating the need for an additional layer in the automation system.
- The real-time control is preformed using traditional control languages such as IEC 61131-3 so no additional level of training is necessary to maintain the MAS system.

We offer some examples of RA multi-agent systems in the next section, and then in the following section, discuss some of our observations on existing obstacles to widespread adoption of agent-based control systems.

2. APPLICATION EXPERIENCE

Rockwell Automation has investigated agent-based solutions for a number of applications and has implemented agent-based solutions for two specific industrial applications. The common requirement of these applications is the need for flexibility and reconfiguration. The justification for these projects has been either the increased utilization of manufacturing assets or a more robust system that can continue to operate during major disturbances.

2.1 Rod Steel Production.

The first RA industrial agent project was to increase the machine utilization of a steel rod bar mill. The rod mill is shown in Figure 3.

The mill makes steel rods by reheating steel and rolling the steel to size using multiple rolling stands and cooling the steel along a defined temperature profile using multiple cooling boxes as shown in Figure 4. The production process recipes for most of the steel rods require the use of neither all of the rolling stands nor all of the cooling boxes.

Hence the system had built-in redundancy and flexibility since it could use any combination of cooling boxes and rolling stands from the subset of working units to produce a given steel rod recipe, as long as the required temperature profile was followed. The recipes that the operators had been using, however, specified particular subsets of cooling boxes and rolling stands. If some piece of specified equipment was broken, the operators would not run the recipe, and the order could not be filled. The aggregate desired behavior of the MAS was to select and configure a subset of cooling boxes from the working units to satisfy the recipe requirements. This was implemented by enabling each cooling box or unit to assess its own health and bid on its part of the operation. The bids were used with a very accurate simulation of the steel cooling process to enable rebidding until a suitable subset of units and configurations was found as presented by Vasko (2000).



Fig. 3. Bar steel production



Fig. 4. Bar steel mill process diagram

The process used to develop the agents for each of the participating units was to first understand the requirements. The operators were interviewed to understand what aspects were important to control the quality of the steel. Once the requirements and the priority of the requirements were understood we began to investigate various proposed solutions. Using the agent concept we assigned functionality to the devices in the system. The cooling box agent was given the task of determining its health and capacity to cool by maintaining a history of water flow. A central node contained the steel cooling simulation that arbitrated the bids to obtain the correct cooling curve. A simulation of the entire process was used to evaluate the agent systems performance. Changes were made to the behaviors of each agent until the overall desired behavior was achieved.

The agent-based control system did not directly control the bar mill but instead recommended a configuration to the operator. Because of safety concerns and possible damage to equipment the risk was too high to enable direct control by this new technology. Although the agent system preformed very well in all the tests, to release the system for production would require testing all steel recipes with all possible subsets of cooling boxes.

2.2 Navy Chilled Water System

The Office of Naval Research (ONR) was looking for a highly survivable robust control system for the chilled water distribution system, one of the critical ship systems. One of the major requirements was that the chilled water system continues operation even after a major disturbance such as a missile strike or explosion somewhere on the ship. Several approaches were investigated and a distributed intelligent multi-agent-based system was selected. The main goal was to have a fully distributed system with no single point of failure.

In a conventional chilled water system one controller with distributed I/O would suffice to handle the normal operation of the system. By normal operation we mean to control the valves and chillers to regulate the circulation of cold and hot water. This single controller case is what we refer to as monolithic control. The monolithic control offers a solution that contains a single point of failure and is difficult to expand or maintain as the requirements evolve. If the system is distributed among a number of controllers to increase survivability the tight coupling between controllers may not yield a more robust system and these systems tend to be even more difficult to maintain.

The RA solution was to use the agents consisting of both reasoning and real-time control, distributed among 23 controllers which were physically located near the control hardware. The reasoning part of the agents inside the controllers negotiates the control actions. The intelligent agent control is a good

alternative solution to monolithic control because it is distributed but the nodes are only loosely coupled. Each intelligent agent provides actions to handle the normal operations but also adds diagnostics and system reconfiguration upon failure. The beauty of this approach rests on the emergence of the solution. As opposed to the centralized case, the agents discover what to do when a portion of the system is no longer operational. In the centralized case, the control engineer needs to program every possible combination of failures ahead of time. This is an NP-hard computing activity, which is difficult and expensive to program at best.

In Figure 5, the Navy water cooling system schematic is shown. This cooling system is a land-based scale model used for Navy testing purposes. It includes the plumbing, controls and communications, and electrical components that mimic the real-life operations. A typical plan consists of water routes to transport cold water from the coolers to the loads and water routes to move hot water from the loads back to the coolers. The hot and cold water plans, which do not have to transverse the same route, combine to form a complete water route to cool a specific load. The agents evaluate a number of real-time conditions in selecting a water route. Each agent evaluates the physical condition of the equipment under its control to decide its participation in the water routing plan.

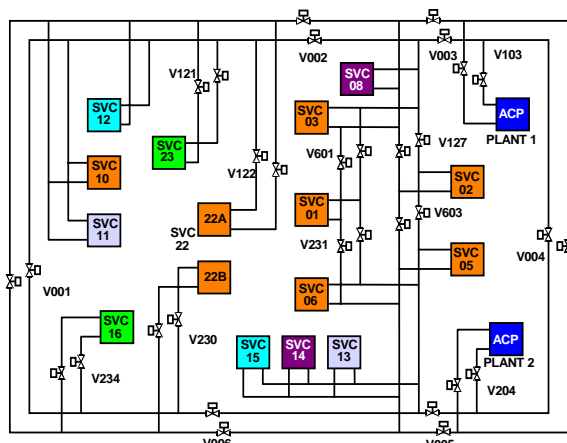


Fig. 5. Schematic of chilled water test system

Immersion heaters provide stimuli for each service (SVC boxes in Figure 5) so as to provide actual heat transfer. Essentially, there are 3 subsystems: plants, mains and services. There is one plant per zone (i.e., currently 2 plants: ACP boxes). There are two types of services, vital (14) and non-vital (2). While in operation under normal conditions, the cooling system is segregated in two zones to maintain the cold water from each source separate. These two zones increase the survivability of the system in case of damage occurring on one side. It is a requirement that the water from the cooling plants never be mixed.

To facilitate experimentation and development of the behaviors for the agents we built a “Table Top” physical simulator shown in Figure 6. This prototype provided information on unknown behaviors such as

the directions of the water flow inside the pipes for specific valve configurations.

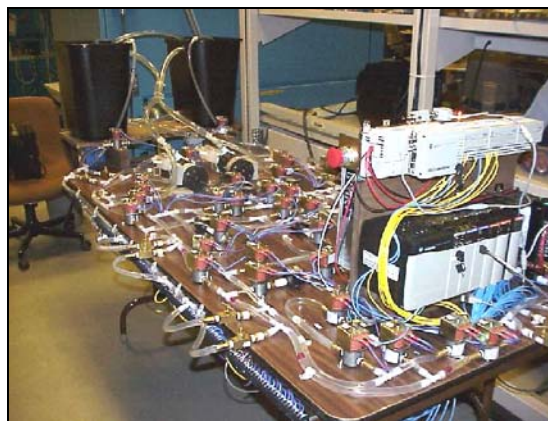


Fig. 6. Physical prototype of the chilled water test system

Our first attempt at an agent-based solution was a very centralized algorithm that required knowledge of the state of each valve and performed the routing in a single agent. Although the algorithm functioned it was not very robust and failed the “no single point of failure” test.

The second attempt at a solution began with a more distributed approach which assigned segments of the algorithm to various agents. This did yield a system which met the no single point of failure test but due to the number of messages generated could take a very long time to find an acceptable route.

The third attempt uses a methodology similar to “responsibility driven design” as presented by (Wirfs-Brock, *et al.*, 1990) from the object-oriented world. We looked at a single valve and asked the question: “what is the minimum information a valve needs to decide to open or close?” From this we developed a simple set of rules for a valve. The same set of rules was applied to each valve in the system. We used the same methodology for the chillers and the loads. The resulting system had very few total rules and the same rules for each agent of a particular device type. The system was simpler than any of the previous versions and was easily scalable. Larger systems contained more instances of the same agents with the same simple rules. The development of the rules for each device was an incremental process. The agent development environment (Staron, *et al.*, 2004) enabled developing a new set of rules or agent scripts as templates; the templates were used to instantiate a set of agents for the devices. These agents were loaded into the controllers for the “Table Top” demonstrator and the aggregate system behavior observed. Based on the system behavior we modified rules or added new rules to move the system closer to the desired overall behavior.

The development of the agent control system was progressing but at a slow pace since we had only one “Table Top” simulator that constantly needed maintenance to fix problems such as leaks, stuck valves, and clogged pipes. A better method was

necessary, and we decided to build a software simulation.

Building a simulation model of closed water circulation is a nontrivial task. We needed to model the water flow and heat transfer properties of a system that already existed. Moreover, our simulation needed to have the same input/output interface that the real system did, so that our agent-based control system could easily control both.

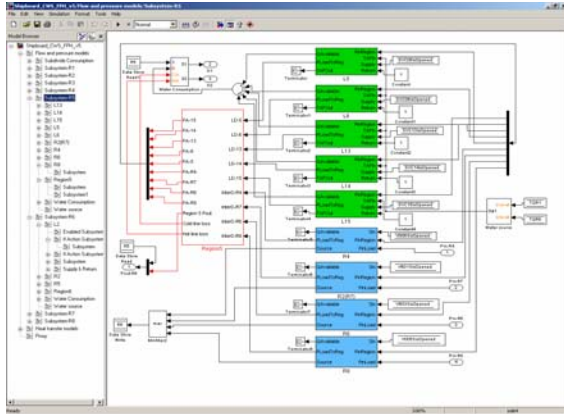


Fig. 7. Simulink simulation of the chilled water test system

The simulation was too complex to run in real-time so we synchronized the clocks so that time passed at the same rate for both the control system and the simulation. We not only modeled the physical properties of the plant but also added if-then scenarios (such as broken components and leakages) to further stimulate the intelligent agents and the control system.

We built a Simulink 6.5.1 simulation, shown in Figure 7, of the circulation system to run on Windows XP. The simulation was synchronized with 3 soft controllers (SoftLogix 5800 from Rockwell Automation) running on the same computer. The controllers were programmed with relay ladder logic IEC-61131-3 programs and 68 agents were created to represent the equipment and sub-systems.

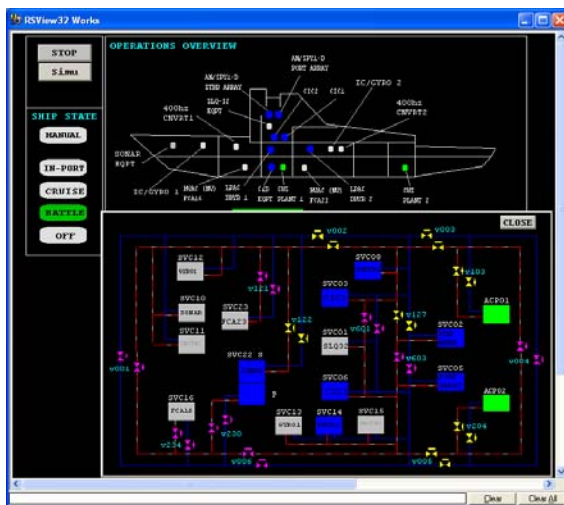


Figure 8: Operator interface

To observe the aggregate behavior of the system an operator interface was built using standard industrial display software, Rockwell Automation RSVIEW32. Figure 8 shows an operator interface view of the shipboard system and its chilled water system operating under battle conditions.

With the software simulation and the graphical display of system status multiple people performed incremental improvements to the rules for the individual agents and observed the results. The software simulation also made the system easier to demonstrate, since both the simulation and the agent-based control system can be operated on a single computer.

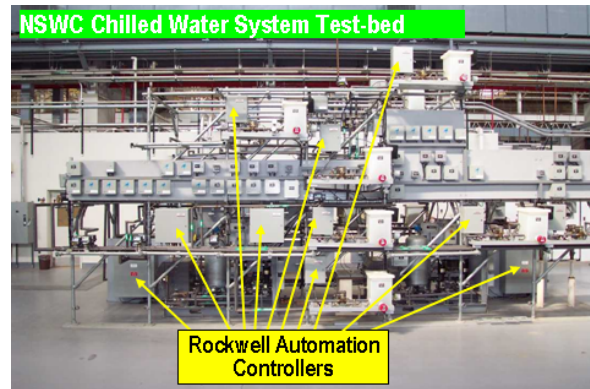


Fig. 9. Office of Naval Research chilled water land-based simulator

The completed system was simulated in software and then deployed and demonstrated on the ONR land-based simulator shown in Figure 9. RA is continuing its activities on agent-based control systems, but only a relative few control system customers have demonstrated a real interest. The next section offers some observations regarding some barriers RA sees to widespread adoption.

3. OBSTACLES TO WIDESPREAD ADOPTION

The major obstacle to agent adoption is the risk of a successful project versus the long term payback. The risks for an agent-based system are:

- Can an agent-based system be designed to meet the system requirements?
- Can the aggregate behavior of the agent-based system be guaranteed to meet all the system requirements?
- Can the agent-based system be cost effective compared to a more traditional approach?
- Can an agent-based system be operated and maintained by existing factory personnel?

These questions will be investigated in more detail in the following sections.

3.1 Lack of Skill in “Distributed” Thinking

The agents within a system are each programmed with a set of simple rules so that their aggregate behavior yields the desired system behavior. The root design philosophy is that of object-oriented technology, i.e., programming a distributed system of

agents or objects. This “distributed” thinking about control algorithms is difficult to master. Our education system, for example, trains its engineers basically to consider algorithms in a centralized system, i.e., running from start to end on a single computer. There are very few courses in distributed problem solving although books on the subject do exist, for example (Resnick, 1997). RA is working with a number of local and international universities to establish courses in distributed problem solving.

3.2 Determining Emergent Behavior

In designing an agent-based system, a designer typically starts with a set of system requirements. But there is no known formal procedure or algorithm to transform those requirements into the small sets of rules for each of the agents in the distributed system. Nature, on which agents are based, may take millions of years and try all possible combinations, discarding all that fail to survive to evolve the rules for agent behavior. This method, although effective, is not practical in industrial automation. We need research into methodologies such as “Responsibility driven design” to enable “evolution” of agent-based systems in a more practical time. Likewise, given the sets of rules with which each of the agents have been programmed, there is no formal procedure or algorithm to generate the set of resultant system behaviors. Thus there is no guarantee that the system will function as desired. RA has used simulation of the machine or process successfully to view and study the control system’s emergent behavior, though building the simulation is at least as much design and implementation effort as building the agent-based control system.

We have observed the emergence of design patterns, similar to design patterns in object-oriented design, as we have studied the use of agents in a number of applications. RA, in its agent development environment, employs a template library to remember the design of each class of agent. Each library is application specific, so that each library can remember a set of reusable agent design patterns. We have, for example, a library for chilled water systems that contain design patterns for routing through a network. These “routing” designs might be reused to route electrical power or packages on a set of conveyors. The use of these patterns increases the confidence that an agent-based control system will exhibit the proper emergent behaviors. This use of agent design patterns is an area of continued study and research.

3.3 Cost of Adoption and Implementation

As mentioned above, for a system to benefit from an agent-based control system, there must be some decisions for the agents to make, i.e., there must be some redundancy or flexibility in the system itself. The capital cost, for example, of adding a more flexible material handling system or redundant equipment to an existing system, thus enabling the agents to be useful, may be prohibitive. This cost

may explain why agent technology in the information technology sector, which requires little additional capital expenditure, has been adopted at a much faster rate. The additional hardware costs of an agent-based control system are negligible compared to the capital cost of machinery, but there are some additional software development costs for the implementation and testing of agents.

3.4 Maintainability of Agent-based Systems

Most industrial automation systems are expected to last in excess of ten years. A major concern is the cost of training and maintenance of any system in a factory. RA chose the IEC 61131-3 languages for the real-time control aspects of the agents because that is where the majority of the routine maintenance is required. The great majority of our customers’ personnel are familiar with those languages and therefore some of the training costs can be kept low. Expanding an agent-based system may only require more instances of the same agents. But enhancing the behavior of the system requires control engineers skilled in the art of agent design. This emphasizes the need for the educational requirements stated above.

Lastly, the controls marketplace has traditionally been a conservative one. Most installations must function for many years, and designers want proven technology supported by multiple reliable vendors that have adopted widely used standards.

4. SUMMARY

In this paper we have presented a summary of Rockwell Automation’s agent philosophy and work to date on agent-based control systems. Effort within RA continues in the areas of developing more design patterns and application libraries, improving our agent design and simulation tools, using emerging agent related standards, and convincing customers that intelligent agents offer solutions to some of their current and future control problems.

REFERENCES

- IEC (International Electrotechnical Commission) (2001), TC65/WG6, 61131-3, 2nd Ed., Programmable Controllers - Programming Languages
- Koestler, A., (1967) *The Ghost in the Machine*, Arkana, London
- Resnick, M., (1997) *Turtles, Termites, and Traffic Jams*, MIT Press, Cambridge, Massachusetts
- Staron R., Maturana F.P., Tichý P., and Šlechta P. (2004) : Use of an Agent Type Library for the Design and Implementation of Highly Flexible Control Systems. *8th World Multiconference on Systemics, Cybernetics, and Informatics, SCI2004*, Orlando, FL.
- Vasko D., Maturana F., Bowles A., and Vandenberg (2000) : Autonomous Cooperative Systems Factory Control. *PRIMA 2000*, Australia, 2000
- Wirfs-Brock R., Wilkerson B., and Wiener L. (1990) *Designing Object-Oriented Software*, Prentice-Hall