Towards a Multi-Agent Architecture for Process
Supervision and Control System

J. F. P. Queiroz1, I. R. Guilherme1, and R. L. Cagnin1
1DEM/IGCE, UNESP – Univ Estadual Paulista, Rio Claro, São Paulo, Brazil

Abstract - Technological advances have leveraged the use of devices in industrial environments, which have contributed to the increase in their complexity. In this sense, there is a need for new technologies and approaches to address involved complexity and develop computational systems that provide tools to integrate heterogeneous components, manage and analyze the huge amount of data produced, in order to automate the activities of process supervision and control. In this context, the approach of multi-agent systems, service-oriented architecture and semantic technologies offer promising and novel ways to design and develop complex, distributed, flexible and scalable systems. This work presents a multi-agent architecture based on these technologies aiming to provide a reference model for the development of process supervision and control system. The proposed architecture was specified based on a literature survey and brings together the main requirements and features related to the use of these technologies for process supervision and monitoring.

Keywords: Multi-agent System; Semantic Web; Service-Oriented Architecture; Supervision and Control System

1 Introduction

The technological advances in software and hardware have caused a cost decrease and consequently an increase on the industrial automation and information systems. With the increased use of controllers and measurement devices, the amount of generated data on the industrial environment has also suffered this effect. Many of the produced data are simply stored and underused or wasted, mainly because it is necessary a great effort, time demand and specialized knowledge for their integration and analysis. Most often, they represent an important information resource that can be used to improve the processes management and control.

In this context, new technological approaches based on computer systems need to be studied and proposed for processes supervision and control, not only for retrieving and analyzing the large volume of data, but also to automate other tasks related to the process monitoring and control support, assisting engineers and operators in decision making.

Such software systems need to deal with the complexity inherent to industrial environments, characterized primarily by the distribution and amount of heterogeneous components, as well as the large volume of produced data. This problem arises from the different communication technologies and interfaces used by these components. Therefore, the integration and management of processes and components are essential and a major requirement for supervisory and control systems. Other characteristics such as flexibility and scalability are also important, because these environments often require changes by updating or adding new resources such as devices and software modules [1].

The design and development of systems with these features need to be based on robust architectures and at the same time flexible enough to support system and environment upgrades and changes. Also they should provide tools for automating some user tasks and support them in decision making, improving their supervision and control capabilities.

In this sense, the multi-agent systems (MAS) approach provides more appropriate abstractions to deal with complex and distributed problems [2]. It aims to simplify the understanding, design, development and management of distributed, open, dynamic and heterogeneous computer systems [3]. Also in this context, web technologies such as service-oriented architectures (SOA) have been adopted to support the interoperability among heterogeneous components and systems. Ontologies have also been used to model the application knowledge and to provide better data integration and processing.

The goal of this work is to describe an architecture model that integrates the MAS approach, SOA and ontologies in order to be used as basis for the design and development of more open, flexible and scalable software solutions for the process supervision and control in industrial environments. In this sense, we present a multi-agent architecture defined and specified based on a survey of a set of related works that have presented approaches in this domain. Therefore, the proposed architecture aims to bring together the main requirements and features of supervisory and control systems. It focuses mainly in the process supervision and monitoring aspects related to data acquisition, integration and processing, in order to automate tasks and support users with relevant information in abnormalities prediction, detection, diagnostics and recovery.
2 Related work

Several approaches reported in the literature apply the agent-based technology in different industrial domains covering manufacturing [4] and continuous process industry [6]. The MAS approaches have been used not only in academic research, but also in industrial projects [4-6], to solve problems ranging from the integration of heterogeneous and distributed data and components [7-8], to the automation of monitoring and control of processes and environments, in order to identify and recover from the occurrence of abnormalities [9-11].

As a complementary solution to agent-based technology, some approaches have also proposed the use of SOA [12-13], as well as semantic web technologies such as ontologies [7, 9, and 14]. Service-based architectures can further improve the system flexibility and interoperability, while ontologies allow semantic annotation of data in order to make it machine readable and interpretable also contributing to automate data searching, accessing, integration and processing.

Based on a literature survey a set of works which uses the MAS, SOA and semantic web technologies to develop supervisory and process control systems was analyzed. In these approaches were identified and analyzed the main requirements and features, which have been used as basis for the definition and specification of the agents and the infrastructure concerning the proposed MAS architecture. The following are discussed the main features identified.

2.1 Components and data integration

The integration of components (devices and legacy systems) and data is one of the primary requirements in complex environments, characterized by several distributed heterogeneous components. On such environments, the majority of components have their own interfaces which are generally incompatible with each other and work with different communication protocols and standards, contributing to the wide variety of data models and formats.

The agent-based approaches have been used to integrate and manage the different components and the produced data in order to improve and automate the execution of process and analysis tasks. This integration can be done at low-levels, among components, and at high-levels between components and high-levels systems. In high-levels ontologies have also been employed to further improve the integration and processing of data by machines.

It has been identified two main solutions which lead to the components integration: (1) agents implementing the required communication interfaces to access their resources and services, (2) adoption of service-oriented architectures, such as web services, where components are wrapped and provide their resources through standardized interfaces that can be easily accessed by agents and other systems. In both cases the data will be provided by standardized interfaces and can be semantically annotated, allowing agents to find and consume these data dynamically and automatically, thus minimizing the human intervention.

2.2 Monitoring and data analysis

The monitoring and data analysis is a feature related to the capability to detect, find, diagnose and predict the occurrence of abnormalities, by continuous monitoring and analysis of process measured variables. The monitoring process can be embedded into devices, if they support resource constraints, or by high-level monitoring systems. In the surveyed approaches were identified a set of agents employed to perform monitoring tasks. They can also perform the pre-processing and data analysis.

2.3 Planning and control

This feature is related to the management of the processes, comprising the tasks to determine and execute control plans, assessment and simulation of control actions, avoid and recover from abnormalities, and adapt to environmental changes. In the surveyed approaches these activities are performed by a set of agents who have knowledge about the processes and often capabilities for cooperation and negotiation with other agents.

2.4 User interface

This feature comprises the supervisory and control system GUIs, responsible for present to the users the functionalities and the resources of the application. These interfaces can be customized and developed with different technologies. Through the GUIs users can interact with the application in order to monitor and control all the related processes. Many approaches use a personal agent for managing these GUIs, which makes the interface between the users and the other agents of the MAS, retrieving and integrating the information in a customized manner.

2.5 Agents interaction

Another aspect identified is the interaction between agents that is related to the way they communicate with each other to negotiate and exchange resources and services. In this context, the use of an agent communication language and common protocols are responsible for ensuring the agents inter-operation. The agent communication language defines the message structure while a shared vocabulary describes the message content. The vocabulary is usually defined by ontologies, used to represent the application domain concepts in such a way the message content can be understood by agents.
2.6 Multi-agents system organization

The system organization is related to the MAS architecture, i.e. the way the agents are distributed, the interaction among them and also between external components. In the surveyed approaches have been identified two types of MAS organization.

In the first approach, the agents do not have an explicit organization, instead they have specific roles and they are related according to their capabilities. In order to agents find and interact with each other, a widely used resource consists in a broker agent that provides a way for agents to register and search other agents which offers required resources and services. Thus, in these approaches all the application agents just need to know this agent.

In the other approach, the agents are organized hierarchically by layers. Each layer determines the roles played by the agents and hierarchical relationship among them. In this organization, the agents of the lower layer always offer services to the agents of the upper layers. In the supervision and control scenario, the agents in lower layers are responsible for data acquisition, in the intermediate layer they perform the data analysis and processing, and in the upper layer they present the results to the users.

3 Multi-agent architecture

The MAS architecture (Figure 1) comprises a set of agents organized in seven different roles that define their behaviors and capabilities for the management and automation of the processes supervision and control activities. Depending on the need and complexity of the application there can be multiple agents performing the same role, where each one is responsible for specific activities related to its role. These roles and their behaviors are presented in detail in Section 4.

The Figure 1 shows how the multi-agent architecture can be applied in a distributed industrial environment in order to realize the supervision and control of the processes. In Figure 1, lower layers comprise mainly existing physical devices, such as the process instrumentation (sensors, actuators and controllers), data acquisition or providers systems, control systems, and other systems used to retrieve data and control the process. All these components are distributed along industrial environments and can be accessible through standardized interfaces, such as those defined by web services (e.g. following industrial standards as OPC-UA and DPWS [15]), or other specific proprietary interfaces.

Above the physical layer are the supervisory and control systems, deployed in local operational units and often connected to physical devices by a local network. In the architecture these systems comprise a set of agents. The agents can be physically distributed along the industrial environment, deployed on local or remote operational units. The Resource and Controller agents can also be deployed on monitoring and control devices if they support the resource constraints. In the same way, the User Interface agents can be deployed on user mobile devices in a manner they can connect and access the MAS of different operational units at any time through a local or even remote network connection.

Each local operational unit can have an application database (DB) to store the process historical data, retrieved from measurement devices, and information about supervision and control activities. The central remote operation unit can has a central database that keeps and integrates data from local units.

In the same way, each local unit can keeps a knowledge base (KB) responsible for store all the application knowledge, comprising the information about the processes such as devices, operational parameters, abnormalities that may occur, and also information about the agents and their capabilities. All this knowledge is defined by experts in the application domain and it is used by the agents to perform their tasks.

Ontologies can be used to support the semantic representation and sharing of the application knowledge. In this sense, domain ontologies should be specified to define a common vocabulary that describes the domain concepts such as devices, data, abnormalities, agents, and their respective properties. Application ontologies also should be defined to describe the processes, monitoring and control plans, supervised abnormalities and diagnostics. There are already several ontologies developed for different domains that can be used and adapted for this purpose.

![Multi-agent architecture applied in a distributed application.](image)
4 Agents specification

In the proposed multi-agent architecture, agents can perform seven roles, defined based on surveyed approaches. The roles determine the behaviors and capabilities of the agents (Figure 2). The same role can be performed by one or more agents. The following are described each defined role.

4.1 Resource

The main function that characterizes this role is the capability to manage the application data resources (historical and real-time measured process data), and make them available to the other agents of the MAS. Thus, agents that perform this role are responsible for retrieve, integrate and publish the data, hiding the complexity of the communication with the various components to the rest of the MAS. In addition, they must know the application metadata model, and use it to semantically annotate the data through domain ontologies.

In an application there can be several agents performing this role, each one responsible for provide access to a particular data resource or a set of specific data resources.

4.2 Monitor

The main function that characterizes this role is the ability to continually analyze the process data, in order to determine the process or devices states and conditions, identifying or predicting the occurrence of abnormalities. Agents that perform this role must also generate alarms and notifying other agents about the detection of unwanted conditions that can bring the process to a critical situation which need to be treated.

The monitoring activity consists in the analysis of process variables, measured by sensors and retrieved by Resource agents. The method of analysis ranges from simple check of the variation of the measured values to the use of more complex methods, such as machine learning to find patterns in the behavior of these variables. These methods comprise algorithms which may be implemented by the agent itself through the use of third party code libraries. For data analysis can also be necessary techniques for pre-processing, feature extraction, prediction or noisy data avoidance.

In an application there can be several agents performing this role, each one responsible for monitoring one or a set of different process operations or devices in order to identify or predict their conditions.

4.3 Controller

The main function that characterizes this role is the capability to perform control actions on the process to modify its operational conditions. These agents can control devices directly through their access interfaces or through control systems. For this purpose, like Resource agents they should implement the communications interfaces with such systems and devices. In an application there can be several agents performing this role.

Agents performing this role can operate as device controllers, maintaining parameter values inside the set points, or just performing changes in the settings of the control parameters or even executing specified control actions. These actions can be organized into control plans, where each plan is defined for a given situation, in such a way that these agents are able to follow and execute the plans when needed or even through user requests. The control can be executed automatically when there aren’t security constraints, otherwise control actions will be executed only under user request or agreement.

4.4 Supervisor

The main function that characterizes this role is the capability to coordinate Resource, Monitor and Controller agents in carrying out the tasks of supervision and control of processes. The agent that performs this role should be able to carry the necessary tasks to initialize and maintain the agents working properly, even when faced with changes in the environment.

The capabilities defined by this role also comprises integrating partial information about the abnormalities identified by Monitor agents, which monitoring different operations or devices in the process, using different analysis methods for monitoring the same or related operations. Under these conditions the Supervisor agent is responsible for consolidating the information from these agents and diagnosing the abnormalities. The diagnostics should be reported to the User Interface agents, and must provide decision support information such as the probable causes, consequences and corrective measures. Based on abnormalities diagnostics Supervisor agent should have
capabilities to determine the recovery plans and count- 
measures. The recovery plans can be executed through 
requests to the Controller agents.

Depending on the complexity of the application, a single 
agent with this role is sufficient to manage the entire process. 
In other cases there can be several Supervisor agents 
responsible for managing a set of related processes or 
different operational units. In this case, Supervisor agents 
should have capabilities to negotiate between them in order to 
determine the local control actions or plans to be executed. 
This negotiation is necessary because in many cases the 
supervised processes or operational unit are related to each 
other in such way that a local action could affect other related 
or subsequent process. They also could interact between them 
to diagnose abnormalities related to other processes or when 
local knowledge is not enough.

Before execute any change in the process, this agent can also 
have the capabilities to simulate and assess the consequences 
and impacts of these actions. Thus, it can assist engineers in 
decision making.

4.5 User Interface

The main function that characterizes this role is the 
capability to retrieve, integrate and present the processes 
information to the users, concerning their interests. Agents 
that perform this role operate as an interface between the 
users and the MAS, interacting with other agents to answer 
the user requests and providing users with information about 
supervised processes.

For each new user in the application an agent must be created 
to manage its respective profile and preferences, related to the 
supervision and control process activities, presenting 
information and notifying the users according to their 
interests, acting in a personalized way and reducing the 
information overload.

4.6 Knowledge Manager

The main function that characterizes this role is the 
capability to manage the knowledge used in the application. 
Agents performing this role are responsible for managing the 
application knowledge bases, providing access to other 
agents. For this purpose, these agents must implement the 
inference mechanisms necessary to manipulate this 
knowledge. This agent can also monitor changes on the stored 
knowledge, for example, updates or inclusion of information 
about monitoring settings or new abnormalities. The changes 
should be reported to the Supervisor agents to perform the 
necessary actions to adjust the application settings to the new 
conditions.

Depending on the application size and distribution there can 
be various agents performing this role, each one responsible 
for managing the local knowledge.

4.7 Broker

The main function that characterizes this role is the 
capability to manage a registry of the services performed by 
agents in the MAS, in such a way that services can be found 
dynamically by other agents. So, the Broker agent must 
provide mechanisms for agents publish their services and 
search for other agents which provide required services. It 
means that all application agents must also have capabilities 
to register and search for services in the registry of Broker 
agents. The Broker agent must allows the agents of the MAS 
to find and consume, at runtime, a service provided by 
another agent without have to be directly bounded to this one, 
improving the system openness and flexibility. The services 
search and discovery can be improved through the use of 
ontologies to semantically describe them.

Depending on the size of the application and the distribution 
of the agents in different platforms, there can be more than 
one Broker agent. The Broker agent can also have an active 
behavior, reporting other agents about updates in the 
registered services and also about new agents and services in 
the MAS. This behavior contributes to the adaptation of 
agents to cope with changes that may occur in the MAS.

5 MAS organization

The MAS social organization determines the 
relationships between agents that perform the roles specified 
in the previous section. The agents interact to accomplish 
their tasks through messages for exchanging information and 
resources.

5.1 Agents interaction

The relationships among the agents are shown in Figure 3. In this figure the Broker agent does not appear related to 
any other agents for clarity, although it has relationships with 
all other agents of the MAS. All the application agents can 
register and search services through a Broker agent. 
Therefore, the messages exchanged between the Broker and 
other agents may be a request to register or search a service, 
or a response informing the list of agents that provide the 
sought service. In case of Broker agent has an active behavior, 
other agents can also subscribe for it to be informed about 
new agents that provide services and resources of interest.
The User Interface agents interact with all other agents except with Controller. The interaction with the Resource, Monitor and Supervisor agents is basically to retrieve information about the supervised processes concerning user interests such as monitoring data, diagnosis of abnormalities and the decision support information that can help users to manage process, recover or avoid abnormalities. The interaction with the Knowledge Manager is to query or insert the information and knowledge of the processes, abnormalities and agents.

The Supervisor agent interacts with all other agents. The interaction with the Resource agents is to store the records of diagnostics, with the Monitor agents to request abnormalities monitoring tasks and be notified when abnormal situations are identified. The interaction with the Controller comprises requests to perform control actions. The interaction with the Knowledge Manager is to query the knowledge base to retrieve necessary information for the supervision and control of the processes, and also be notified when changes occur in application knowledge. The Supervisor agent can also interact with other Supervisors to negotiate control plans or diagnose abnormalities.

The Monitor interacts with the Resource agents to retrieve the data used in the monitoring tasks and also store the records generated during monitoring activities in the historical application database. The other relationships among the agents have already been presented above.

5.2 MAS functional and implementation aspects

In the industrial environments there are different interconnected units responsible for performing a set of operation. Based on this, the proposed MAS can be implemented for each local and remote operational unit. This organization ensures a local autonomy and independence between units for the process supervision and control. It also allows easily integration with new units, components and agents, without have to change the entire application infrastructure. The MAS must be developed and deployed on agent platforms, compliant with wide accepted standards.

In this MAS organization Resource and Controller agents most often performs only reactive tasks that do not require processing performance, making these agents suitable to be embedded into low-level devices. However, the Supervisor, Monitor and Knowledge Manager agents have more proactive and cognitive behaviors requiring more powerful platforms to be deployed. The User Interface agents can be deployed on user mobile devices and through them users can connect to any operational unit to manage the processes, concerning the required privileges. The GUIs can be implemented by these agents or by a separated application that interfaces with them.

In an application, for each MAS deployed on different operational units, there are one Broker agent, in order to allow the other agents to find each other dynamically. The Broker agents should know each other and share their service records or must have mechanisms to search agents registered on other Broker registry. These mechanisms are important because through them the Supervisor and User Interface agents can find and interact with agents from different MAS.

Likewise, for each MAS there are a Knowledge Manager agent to maintain a knowledge base, storing only the local knowledge, required for local process management, or all the application knowledge. Both strategies contribute to the autonomy of each MAS. Thus even during a communication loss among operational units, each MAS can continue working with their local resources. The Knowledge Manager agent of the central control room should synchronize all the application knowledge, in order to provide a global scenario and assist experts to manage the entire industrial environment. The knowledge should be defined and structured by domain experts in the development phases.

New agents can be developed and deployed in the application. When it happens, the new agent need to register their services through a Broker agent, from this moment the agent will be automatically integrated in the MAS, since other agents can find and interact with it. Also considering service-based technologies, such as OPC-UA and DPWS [15], a specific Supervisor agent can be in charge of search for new devices providing services in the network. Thus, when new devices are discovered, this agent can get more information about it with Knowledge Manager agent and initialize a new related agent or request for existing ones to wrap this device resources. For devices or other component that provide both data and control interfaces, only one agent can be deployed to manage it, so this agent will implement both Resource and Controller role capabilities.
6 Conclusions

In this paper was presented a multi-agent architecture that provides a reference model for the development of applications to process supervision and control. The architecture brings together the main requirements and features for process supervision and monitoring. They were identified in a literature survey and used as basis for the specification of the agent roles and MAS organization. Additionally, the architecture is specified in general terms and can be used to develop solutions for different domains.

In the multi-agent architecture the integration of agent-based, SOA and Semantic Web technologies allows the design of a more appropriate infrastructure to deal with the complexity of industrial environments, contributing to the development of flexible, open, dynamic and scalable systems. In this sense, the modularity and autonomy presented by the agents allow better management, control and integration of the various components. Standardized interfaces and mechanisms for publishing and discovering services offered by SOA, allow interoperability, reusability and dynamic integration of heterogeneous components. The representation of the application knowledge through the semantic Web technologies ensures the autonomy of agents to perform their tasks as well as to discover, integrate, and process the data in a seamless and automatic manner.

For future work we intend to develop applications based on the proposed architecture, in order to validate and show how it can be developed and deployed. Further, improve the architecture specification and develop a framework to support the design of applications for process supervision and control.

7 References


