Agent-Based Industrial Environments: State of the Art

Hosny Abbas, Samir Shaheen, Mohammed Amin

Abstract— this paper highlights the state of the art deployments and applications of the agents and multi-agent systems (MAS) technologies within modern industrial environments. It explores and reviews some of the recent multi-agent based industrial applications to identify their advantages and also their common limitations. The key conclusion of this paper is that most of the proposed agent-based industrial applications adopt the traditional basic forms of the agents and multi-agent systems. They only concern the level of individual agents (the system micro level) by starting the development process with the identification of the individual agents' types and the design of behaviors and interactions of each agent in the system without giving attention to the global system structure and organization (the system macro level) as they assume that the required global system functionality will emerge as a result of the lower level agents interactions and behaviors. This type of development approaches is not adequate for engineering complex, open, and highly distributed modern industrial networks (i.e., modern SCADA). Undesirable system level behaviors can be emerged and can impact system performance and reliability. Present and future complex highly distributed real world applications and especially industrial ones such as those related to critical infrastructures (i.e., power grids, water transportation, etc.) management should be tackled from both of their micro and the macro levels in the same time because they continuously evolve and change in an unpredictable and uncertain manner. This paper highlights the problem and suggests the promising solution.

Index Terms— agents, multi-agent systems, industrial environments, agent-based industrial applications, critical infrastructure, SCADA networks

◆

1 INTRODUCTION

Recently, the agent-based approach has been adopted and used to model, design, and develop a wide range of industrial and real life applications such as manufacturing, power systems, process control, etc. Agents are vital in the globalization context, as globalization refers to an inherently distributed world both from geographical and information processing perspectives. What distinguishes the agent-based approach from other traditional approaches is its unique ability to handle simultaneously many challenges of current software applications specially those applications which are highly distributed and their working environments are highly dynamic and uncertain. MAS provide a suitable paradigm for decentralized systems in which autonomous individuals engage in flexible high-level interactions. The Agent-based approach is considered as a new software engineering architectural style for the development of complex, decentralized and open software applications. Further, the agent-based approach offers a new and exciting paradigm for the development of sophisticated programs in dynamic and open environments [1]. Using MAS-based solutions highly reduces complexity and principally designing with agents is easier for business analysts than using any other mainstream technologies, so information-technology practitioners can easily make the transition from object-orientation to agent-orientation. The agent-based approach is definitely effective in the design and concrete realization of industrial-strength software applications [2][3]. Weyns et al. [4] agued that Autonomy, adaptability, robustness, and scalability, are generally considered as key properties of complex distributed systems, and these are exactly the properties that characterize MAS. The body of knowledge developed by the MAS research community is therefore of crucial importance. It is our firm belief that only by integrating the knowledge in a broader setting of software engineering, the fruits of MAS research will find their way to practice [4]. This paper reviews some of the proposed agent-based industrial applications within industrial environments to identify and highlight there advantages and also there limitations. The key goal of this paper is to assess the current practice of applying the agent-based approach for engineering modern industrial networks and and to highlight the limitations of the current agent-based applications.

The remaining of this paper is organized as follows: Section 2 explores the state of the art of the agent technology

---

- *Hosny abbas is currently pursuing doctoral degree program in computers and systems engineering in Assiut Univeristy, Egypt. E-mail: hosnyabbas@aun.edu.eg*
- *Samir Shaheen is currently a professor of computer engineering in Cairo University, Egypt. E-mail: sshaheen@eng.cu.edu.eg*
- *Mohammed Amin is currently a professor of control engineering in Assiut University, Egypt. E-mail: mhamin@aun.edu.eg*

within industrial environments such as process control, manufacturing, power systems, SCADA, etc. Section 3 discusses the pros and cons of the current trends of applying the agent technology within modern industrial environments and identifies a critical problem that can be an obstacle against the development of complex large-scale industrial systems such as modern and future SCADA networks. Section 4 provides a general discussion and identifies the challenges facing engineers and developers of complex large-scale and highly distributed industrial environments such as modern SCADA networks. Finally, Section 5 concludes the paper and highlights the intended future work.

2 THE STATE OF THE ART

The agents and multi-agent technology has been adopted and used to model, design, and develop applications in a wide range of real life domains. We are concerned in this paper with their applications within industrial environments. The explored applications are only those related to industry and critical infrastructures such as process control, manufacturing, power systems, and SCADA networks.

2.1 PROCESS CONTROL

The process control is used to automatically control a process such as chemical, oil refineries; paper/pulp factories, etc. It often uses a network to interconnect sensors, controllers, operator terminals and actuators to automatically control and supervise a physical control process. Fig.1 demonstrates the possibility of deploying agents within the abstract pyramid of Industrial Contol Networks (ICN), which represents the general conceptual model of ICN, and is represented by three abstract layers aligned on top of each other (ignoring the physical complexity of each layer). The layers of the ICN abstract pyramid are (from bottom to top): The device layer, control layer, and information layer. An important question should be asked here: where in the ICN pyramid agents can be better deployed? For the sake of safety and scalability, the deployment of agents is preferred to be in the whole ICN pyramid layers. As in this case there will be no non-agent entities within the ICN and therefore no custom interfaces (agent-device interfaces) need to be considered. Unfortunately, there are some challenges against achieving that. The first challenge is the real-time requirements imposed on the lower-level layers because they are very close to the physical processes (devices), which work in a time-based fashion. Agents are supposed to be autonomous and able to reason before taking actions, that means there will be a time delay before the suitable action is taken (i.e., reasoning time) when a change in a process takes place because sometimes this time delay is undesirable in certain time-critical processes. In other words, the real-time requirements imposed within the lower-level layers necessitate that the computing outcomes be deterministic and time-based and that is not guaranteed by the agent technology. The second challenge is the vendor-dependency, although the agent technology is widely adopted in many fields, still there is a lack of the technology support on the part of control instrumentation vendors. The agent technology is generally not widespread in modern industry (especially in process control) because of the gap found between agents' theories and industrial applications requirements such as real-time constraints [5].

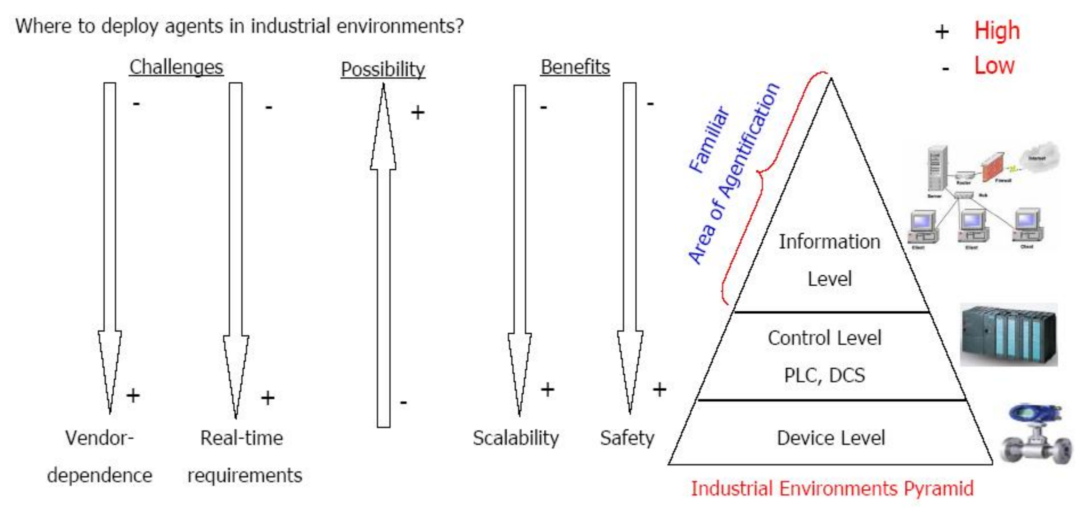


Fig.1 Agents' deployment in industrial environments

The vendors of field devices (sensors, actuators, etc) and control systems (PLC or DCS) are worrying about the adoption of new technology such as the technology of agents within their equipments. To some extent, they may be right because agents are based on the concept of autonomy, which may be a source of the uncertainty. In other words, an individual agent is not reliable and its behaviors are not predictable, so how vendors can depend on its ability to achieve the required functionalities in the suitable time window. Therefore and because of these challenges, the familiar area of agentification is widely located within the information layer (top layer) of the conceptual ICN pyramid shown in Fig.1, where there is independency of the vendors and less real-time requirements.

The authors of [5][6][7][8][9] surveyed the applications of the agent technology in the industrial process control and concluded that the agent technology is particularly popular in the manufacturing domain which comprises a set of discrete processes, while the applications in other domains of industrial control are scarce. That is because when a process automation system is designed, the physiccal phenomena are represented as mathematical models, for which control algorithms are chosen in order to keep the process parameters within a desired range. Therefore, in a single continuous control loop, there is not much place for any additional computational techniques, including the agent technology. In nutshell, agents that are intended to process large amounts of information are better off operating in intermediate and upper levels of the automation stack, where larger blocks of memory and no time-critical operations can be used. On the other hand, agents that are intended to interact with the physical device and whose responsibilities are to steer control and events are better off operating in the controller devices. There has been considerable work on agent systems for the upper levels of control, but very little work has been done to apply these techniques to the lowest control level. To integrate these areas, a simple and expedite interface between the agents and the control functionality is needed [10].

One of the interesting trials to deploy agents inside control devices is the Autonomous Cooperative System (ACS) infrastructure or middleware [11] which aimed at flexible and balanced automation able to survive against contingencies to prevent malfunctioning. Fig.2 shows the location of the agent functionality within the control device which exists as a layer on top of the device's OS. The ACS architecture provides the interface that allows agents to monitor the status of the controller and also helps them in their reasoning about the process. The ACS layer interacts with the controller's firmware to access control functionality such as data table, ladder and communication. In ACS, agent processes are segregated into the lowest priority. ACS has operated as a foreign layer on top of the control firmware. ACS permits to download agent classes into the controllers. The agent classes are instantiated to create specific agents (as instances of a class). Each agent has a control program associated with it. The agents change the control part behavior throughout events and direct access to the data table of the controller. ACS is a task which is initialized during the device's power up cycle. The agents are threads within the ACS environment. These threads are assigned local priorities below the ACS task to avoid interferences with the device's priority. The agent threads are coordinated by a local scheduler. In this configuration, the agent processes are very slow compared to the device's tasks since the ACS task only executes when the controllers is idle from other higher priority activities.

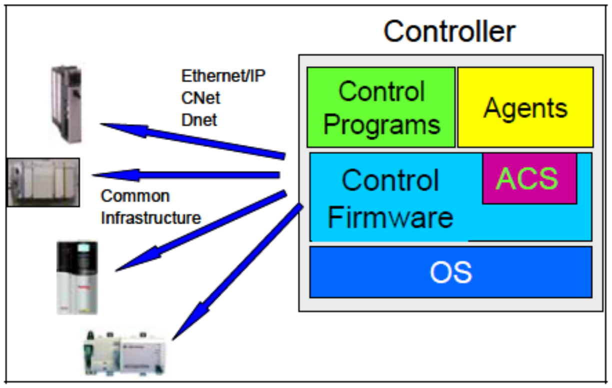


Fig.2 ACS Automation control device.

The ACS approach can be considered as an important step towards enabling control models with agents in an integrated firmware. That will produce a new breed of control devices which can be called as agent control devices. From the other hand, the ACS has obstacles in expandability and upgradeability because it works as a subordinate system separated from the control device firmware. This limitation will make it difficult for the ACS to be formally accepted by the control devices vendors. The authors of the ACS believed that to make a computing unit agent enabled, it is necessary to identify the middleware as a set of functions to be integrated with the device's Operating System (OS). Furthermore, the middleware functions need to be integrated with the hosting OS seamlessly to avoid unnecessary interruptions during the execution of the control tasks. Agent computing is demanding on the CPU and the communication because of the agent-to-agent messaging and reasoning iterations. Hence, to integrate agents within control devices, there is a need to consider the agent functionality as another part of the OS and not as another layer. Currently, there are great efforts to update the ACS architecture aiming to blend the agent functionality into the control firmware. This change means a significant transformation effort and enhancements in relation to the style of information processing for the agent decision making process, handling of messages and events, agent composition and creation, and last but not least software maintenance and upgradeability.

2.2 Manufacturing

In manufacturing, the process consists of discrete and countable components and actions. The natural approach is to assign the software agents to each of the components and each of the actions performed. Manufacturing cycle is the complete cycle of processes which are involved from the very beginning of placing an order to the other extreme of the manufactured product going into the hands of the end consumer [5]. Traditional manufacturing systems were relying upon centralized and hierarchical systems that are not responsive enough to the increasing demand for mass customization. Decentralized, or heterarchical, management systems using autonomous agents promise to nullify the limitations of traditional solutions. Manufacturing systems should be organized as loosely coupled networks of communicating and cooperating components [12]. In manufacturing automation, the delays between the consecutive processes or stages are not vital; usually the process may even be paused for a reasonable amount of time without any negative effect on produced goods. For these reasons, the manufacturing domain is one of the most suitable domains that can be modeled and engineered by the agent-technology due to its discrete nature and its support to decentralized control.

Agents address autonomy and complexity; they are adaptive to changes and disruptions, exhibit intelligence and are distributed in nature. They are well suited to modular problems because they are objects. They are well suited to decentralized problems because they are pro-active objects. These two characteristics combine to make them especially valuable when a problem is likely to change frequently. Modularity permits the system to be modified one piece at time. Decentralization minimizes the impact that changing one module has on the behavior of other modules. Agent-based architectures permit reuse of much existing code and self-configuration of large portions of the system, reducing both the cost and the time needed to bring up a new factory [13]. Shen et al. [14] pointed out that the global competition and rapidly changing customer requirements are forcing major changes in the production styles and configuration of manufacturing organizations. Increasingly, traditional centralized and sequential manufacturing planning, scheduling, and control mechanisms are being found insufficiently flexible to respond to changing production styles and highly dynamic variations in product requirements.

The traditional approaches limit the expandability and reconfiguration capabilities of the manufacturing systems. The traditional centralized hierarchical organization may also result in much of the system being shutdown by a single point of failure, as well as plan fragility and increased response overheads. Agent technology provides a natural way to overcome such problems, and to design and implement distributed intelligent manufacturing environments. There are a large number of applications that adopt the agent technology to manufacturing enterprise integration, supply chain management, manufacturing planning, scheduleing and control, materials handling, etc. For an extensive literature review of these applications see [14]. In this subsection we will only discuss two familiar manufacturing paradigms that adopt the agent technology as an implementation approach: Distributed Intelligent Manufacturing Systems (DIMS) and Holonic Manufacturing Systems (HMS), they have recently been receiving a lot of attention in academia and industry. DIMS, or agent-based manufacturing systems, are based on the agents and multi-agent system technology. The DIMS approach uses heterarchical architectures as the control paradigm. As shown in Fig.3 (which presents the heterarchy structure versus the familiar hierarchy structure), a heterarchical control system is a flat structure composed of independent entities (agents). These agents typically represent resources and/or tasks. Allocation of tasks to resources is done using dynamic market mechanisms. An example application that adopts the DIMS approach is the PABADIS (Plant Automation Based on Distributed Systems) project [15]. The PABADIS deals with plant automation systems in a distributed way using generic mobile and stationary agents and plug-participate facilities within a flat structure as key points of the developed control architecture. It addresses a distributed plant automation system where machines and products are represented by software agents that collaborate to schedule the production and resources in the most optimized way.

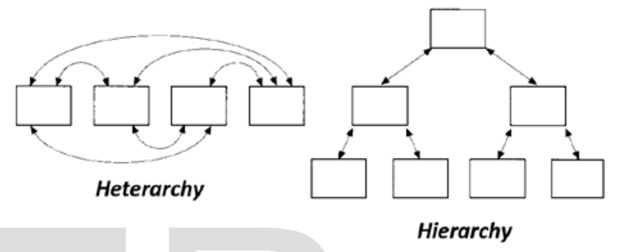


Fig.3 Heterarchy vs. Hierarchy structures

The PABADIS model is based on the following components: CMU (Co-operative Manufacturing Unit), Agents: Product Agent (PA), Residential Agent (RA) and Product Management Agent (PMA), Agency and Look-up service. The combination and collaboration of all of them constructs an architecture whose main objective is to distribute the manufacturing control (MES, Manufacturing Execution System) functions as much as possible within the plant automation system. Fig.4 shows a typical PABADIS model including the ERP System (Enterprise Resource Planning), and several CMU types, all are connected through an Ethernet [229]. A concrete multi-agent system based on the PABADIS model is shown in Fig.5. The figure shows the relationship between the different agents in PABADIS. It is important to note that all PAs work in concurrence to other agents (either other PAs or PMAs) and that their behavior is in strong accordance with common rules.

The PABADIS model, which adopts the DIMS approach, provides product-oriented MAS, which also achieves to decentralize the plant structure. This orientation leads to a more flexible model in both development of production systems and the actual manufacturing process with respect to the individual products.

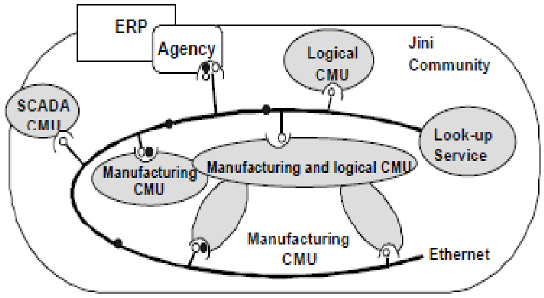


Fig.4 PABADIS model

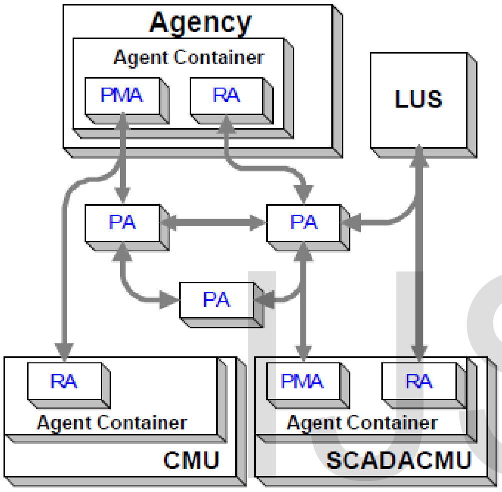


Fig.5 MAS in PABADIS

The DIMS approach has some limitations related to the independence of agents which prohibits the use of global information. Therefore, global system performance is very sensitive to the definition of the market rules (uncertainty); the control system can not guarantee a minimum performance level in the case of unforeseen circumstances; and the prediction of the behavior of individual orders is impossible [16]. The HMS was proposed to solve the DIMS approach problems and challenges. The HMS concept was proposed in 1994 by the HMS consortium as a test case under the international Intelligent Manufacturing Systems (IMS) Research Program [17]. The HMS systems are based on a philosophical concept called as holon. "Holon" is a word coined by combining 'holos' (the whole) and 'on' (a particle) [18]. A holon is defined by the HMS consortium as an autonomous and cooperative building block of a manufacturing system for transforming, transporting, storing and/or validating information and physical objects [19]. Another important concept is the "holarchy" which is defined as "a system of holons which can cooperate to achieve a goal or objective". An HMS is a holarchy which integrates the entire range of manufacturing activities from order booking through design, production and marketing to realize the agile manufacturing enterprise. In an HMS, the activities of a holon are determined through cooperation with other holons, as opposed to being determined by a centralized mechanism. In this type of systems, intelligent agents called 'holons' have a physical part as well as a software part and a holon can be part of another holon. Fig.6 demonstrates how holons form holarchies. The holonic paradigm has a recursive nature and therefore it is best adequate for modeling systems with recursive behaviors and structures, the HMS is one of this systems.

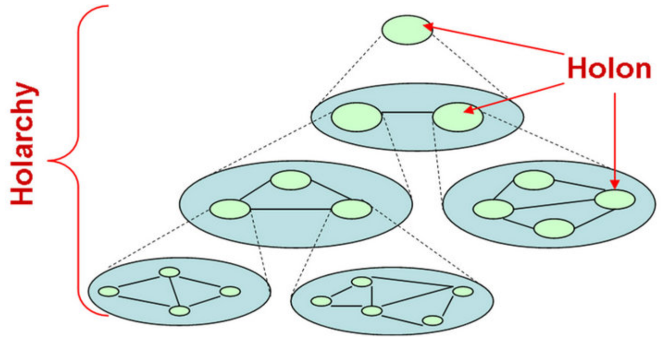


Fig.6 Holons and Holarchies

The HMS concept combines the best features of hierarchical and heterarchical organization [20]. It preserves the stability of hierarchy while providing the dynamic flexibility of heterarchy. An example application of the HMS is the ADACOR (ADAptive holonic COntrol aRchitecture) [21] and the PROSA (Product-Resource-Order-Staff Architecture) [22]. The HMS can be considered as a general paradigm for distributed intelligent manufacturing control, while multi-agent systems are regarded as a software engineering approach for implementing HMS [23]. Although the HMS model achieves to decentralize the plant automation, it already presents a number of disadvantages, (1) Non product-oriented, but rather plant oriented, (2) Not enough flexibility regarding products, because control is done in a centralized or partly distributed way, where the set of agents are performing the control of the whole plant [24].

2.3 Power Systems

Different driving forces push the electricity production towards decentralization. As a result, the current electricity infrastructure is expected to evolve into a network of networks, in which all system parts communicate with each other and influence each other. Electricity transmission networks consist of a number of substations interconnected by transmission lines. Each substation contains transformers, switchgear (disconnecters and circuit breakers) and other items of plant and protective equipment [25]. In the power transmission industry three types of automation system are used: SCADA systems, Energy Management Systems (EMS) and Substation Automation Systems (SAS) [26]. These systems form a hierarchical structure, with the EMS on the top level, the SCADA system directly subordinate to the EMS, and, at the lowest level, the individual SASs.

There have been several applications of the agents and MAS to various areas of power systems. MAS are now being developed for a range of applications including diagnostics, condition monitoring, power system restoration, market simulation, network control and automation [27]. MAS and electronic markets form an appropriate technology needed for control and coordination tasks in the future decentralized electricity networks. PowerMatcher [28] is a market-based control application for supply and demand matching (SDM) in electricity networks with a high share of distributed generation. SDM is concerned with optimally using the possibilities of electricity producing and consuming devices to alter their operation in order to increase the over-all match between electricity production and consumption. PowerMatcher is a market-based control concept for supply and demand matching (SDM) in electricity networks with a high share of distributed generation. In the PowerMatcher method each device is represented by a control agent, which tries to operate the process associated with the device in an economical optimal way. The electricity consumed or produced by the device is bought, respectively sold, by the device agent on an electronic exchange market. The electronic market is implemented in a distributed manner via a tree-structure of so-called SD-Matchers, as depicted in Fig.7.

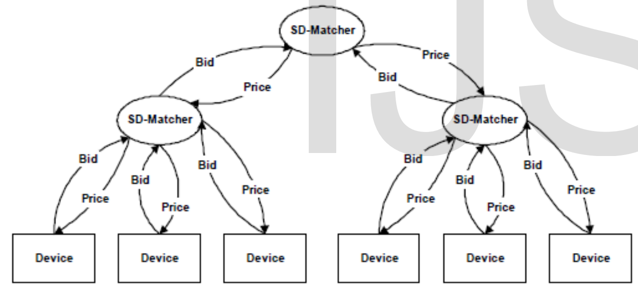
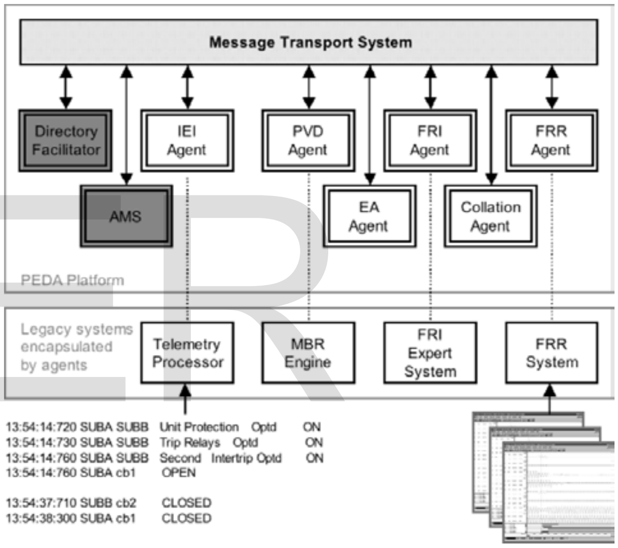


Fig.7 Hierarchy of supply & demand matchers in the PowerMatcher concept, the SD-Matchers implement a distributed electronic market.

An SD-Matcher matches demand and supply of a cluster of devices directly below it. The SD-Matcher in the root of the tree performs the price-forming process; those at intermediate levels aggregate the demand functions of the devices below them. An SD-Matcher cannot tell whether the instances below it are device agents or intermediate SD-Matchers, since the communication interface of these are equal. The pros of the PowerMatcher application are providing distributed rational decision-making, utilizing flexibility in device operation via agent bids on an electronic power market, increased simultaneousness between production and consumption of electricity by devices in a sub-network, and the net import profile of the sub-network is smoothed and peak demand is reduced, which is desired from a distribution network operational viewpoint. From the other hand, it has a number of cons, for example it works as a closed system with static hierarchy and it is scalability is low.

In [29][30] the authors used the multi-agent technology to automate the management and analysis of SCADA and digital fault recorder (DFR) data. The multi-agent system, entitled Protection Engineering Diagnostic Agents (PEDA), integrates legacy intelligent systems that analyze SCADA and DFR data to provide data management and online diagnostic information to protection engineers. The authors claimed that PEDA supports protection engineers by providing access to interpreted power systems data via the corporate intranet within minutes of the data being received. The PEDA architecture is provided in Fig.8 which demonstrates how the PEDA approach can be modeled and implemented using a FIPA compliant agent development platform such as JADE [31]. Similar to an earlier MAS-based application for power systems, called ARCHON [32][33], which concerned the integration of legacy systems found in power systems, PEDA exploits the properties of autonomous intelligent agents to integrate legacy systems in an extensible and flexible manner.



```
13:54:14:720 SUBA SUBB  Unit Protection  Optd      ON
13:54:14:730 SUBA SUBB  Trip Relays  Optd          ON
13:54:14:760 SUBA SUBB  Second  Intertrip Optd     ON
13:54:14:760 SUBA cb1    OPEN

13:54:37:710 SUBB cb2    CLOSED
13:54:38:300 SUBA cb1    CLOSED
```

Fig.8 FIPA agent management reference model, including the PEDA agents and the legacy systems the agents encapsulate.

2.4 SCADA

SCADA networks can be found in power systems, critical infrastructures, space navigation, process control, manufacturing, etc. Thus, SCADA is a general term normally used to describe local or remote supervisory and control activities within a certain industrial domain. SCADA is not a standalone research discipline, but it is the integration of variety of computer science and engineering disciplines. If we explored the history of SCADA, we will find different types of technologies that SCADA adopted to realize local or remote supervisory systems, for example, custom SCADA, component-based SCADA [34], web-based SCADA [35][36], and currently agent-based SCADA. The limitations of one technology impose the transfer to a new technology; therefore SCADA can be described as a technology-dependent discipline. In this subsection we are concerned with the agent-based SCADA.

Many researchers developed and implemented many valuable and feasible agent-based industrial applications. For instance, Diaconescu et al. [37] proposed a concrete approach for linking agents with the industrial equipments (i.e., PLC, DCS, SCADA, and HMI) comprised into a distributed industrial control system based on agents, using OPC servers [38]. Their research tackled the application of the agent technology for monitoring, collecting and archiving data of a manufacturing process in the automotive industry. The contributions of the authors are mainly directed to achieve the connection between JADE agent development framework [31] and process OPC servers, see Fig.9, but they did not exploit the advanced features provided by an agent platform such as JADE, such as ontology support, agents' cooperation, advanced interaction protocols. These features have been proven to be very important especially for open and large scale systems.

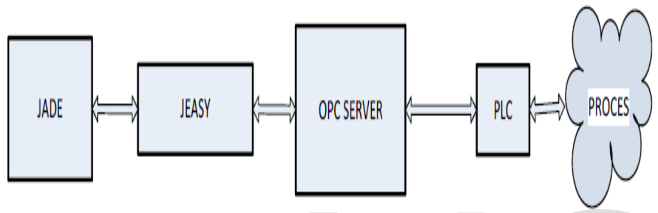


Fig.9 JADE-OPC application structure

Pereira et al. [6] discussed the current challenges of the deployment of MAS in the context of industrial applications, mainly focusing the integration of agents with physical equipment and the ability to run agents directly in industrial or low cost controllers. To support their claims the authors provided an experimental MAS solution for a smart grid case study. The main concern of the authors was how to integrate agents with physical equipment. The authors deployed and implemented their MAS application using the JADE framework. Fig.10 presents the physical smart grid case study they modeled with MAS, and Fig.11 provides the deployment architecture of the proposed application. The communication between system agents and the control systems (PLCs) is based on serial communication.

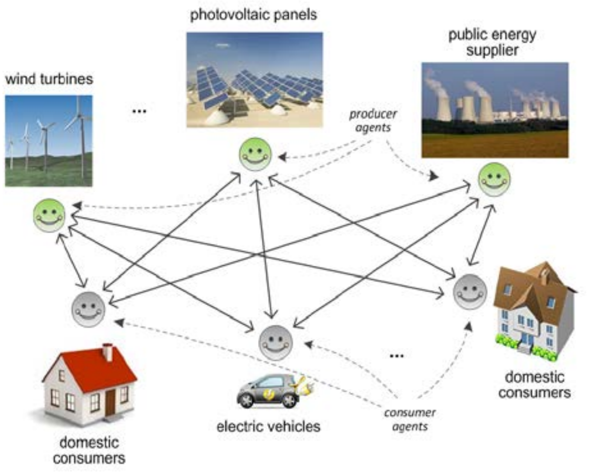


Fig.10 Experimental case study scenario

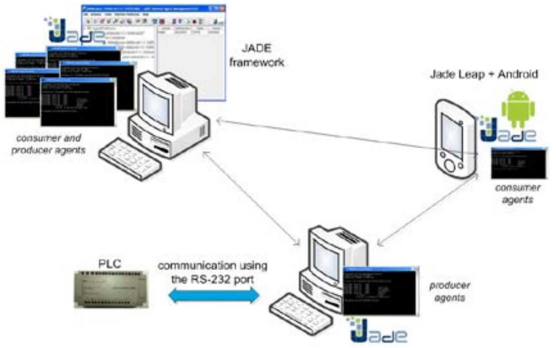


Fig.11 Deployment of agents in a system network

Rupare et al. [39] proposed an automated grinding media charging system incorporating a multi-agent system developed in JADE too. A control logix program is designed to determine the precise quantities of grinding media to be charged in an incremental manner such that shock loading is avoided. The proposed MAS monitors the power drawn and the mill load of the ball mill such that proper charging conditions are established. High quality of the regulation process is achieved through utilization of the control logix and the MAS. Fig.12 presents a typical ball Mill system. Fig.13 and Fig.14 presents the system architecture the agents' structure of Rupare et al. work respectively. According to Rupare, the grinding or milling in ball mills is an important technological process applied to reduce the size of particles which may have different nature and a wide diversity of physical, mechanical and chemical characteristics. As shown in Fig.13 above, the exchange of information and messages is facilitated by the use of an Object Linking and Embedding for Process Control (OPC), wherein the JADE program acts as the OPC server and the Siemens program as the OPC client. The proposed multi-agent system consists of two types of agents, the control agent and the charging agent, each of them has a pre-specified number of responsibilities.

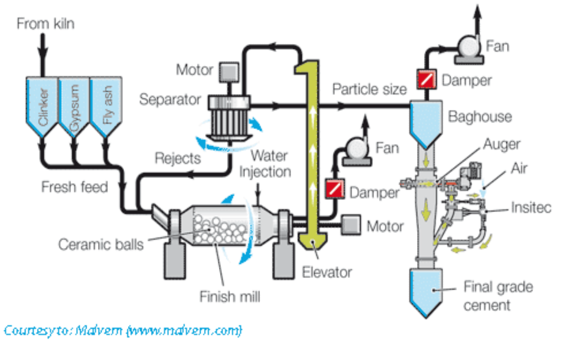


Fig.12 Typical Ball Mill

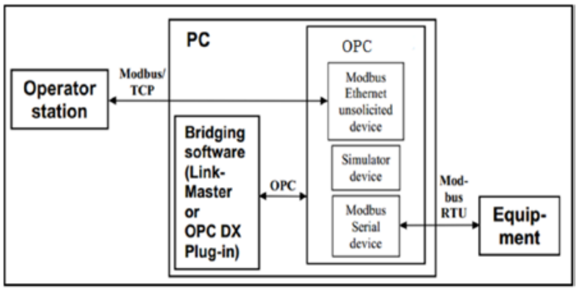


Fig.13 Block Diagram of the Software and Hardware (Rupare et al. [156])

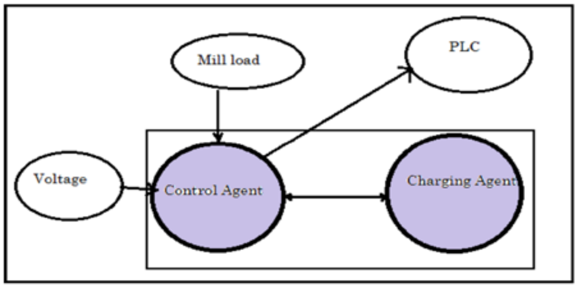


Fig.14 Collaborative Diagram: agents' structure (Rupare et al. [156])

ProVis.Agent [40] is the first agent-based production monitoring & control system for distributed real-time production monitoring. According to the authors of the ProVis.Agent, Production Monitoring and Control (PMC) systems (it is the same as SCADA) play a central role to the classical automation field. The main function of those systems is to gather signals produced by plants and PLCs, combine them to control relevant contexts, visualize them and provide facilities to operate them. The problem with PMC units is the integration between systems coming from different vendors. The authors see that the agent technology presents a more promising technology for integrating existing software systems and their functionalities and to add assistant systems for the shop floor staff. Their viewpoint is demonstrated in Fig.15. Fig.16 shows the use of software agents within the actual system architecture of ProVis.Agent. The central Monitoring & Control Server consists of a collection of cooperating software agents. Each of these agents covers one piece of functionality. It contains the functional treatment of different types of signals (e.g. switches, analog values, distances, etc.) as well as working time models, alarming and statistical data. For example, the visualization agent is used for interfacing with a variety of commonly used SCADA systems (WinCC, FactoryLink, etc.).

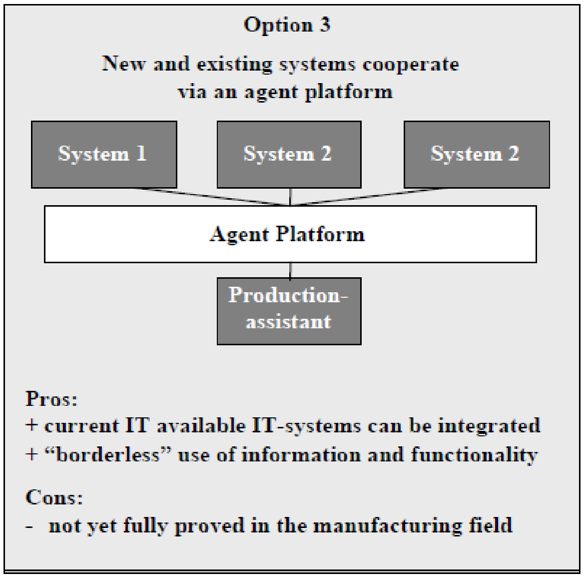


Fig.15 Integration on an agent platform

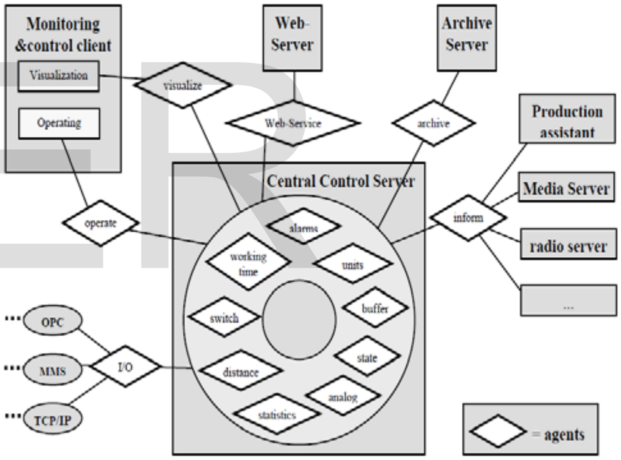


Fig.16 Agent structure of ProVis.Agent

Eleni [41] proposed an intelligent agent-based monitoring system to achieve and advanced condition monitoring for data interpretation of a large volume of data from a Gas Insulated Substation (GIS), which (according to Eleni) has been a successful case study for agent-based data interpretation and monitoring. Fig.17 presents the architecture of the GIS agent-based monitoring systems. The proposed agent-base system consists of a number of agent types, and each agent has predefined responsibilities as shown in the figure.

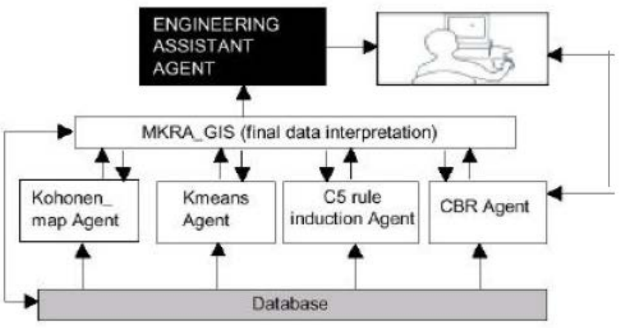


Fig.17 General System architecture for GIS monitoring

Prayurachatuporn et al. [42] concerned the application of software agents to increase the reliability of control systems. They claimed that the software agent technology can improve SCADA systems as it allows distribution, which inherently promotes redundancy, and modularity, which promotes versatility. The authors presented how the development of a directory service protocol using software agent technology increased the theoretical reliability of SCADA. Applying an agent execution environment to real-time distributed control systems means to develop a new protocol for a distributed control system that utilizes mobile agents. An agent execution environment is a software system that provides a runtime environment for agents to execute, a standard interface for interactions, services for creation, migration and termination of mobile agents, supports agent mobility and communication while providing security for both hosts and agents. Fig.18 illustrates how and agent-based SCADA architecture can be according to the authors' viewpoint.

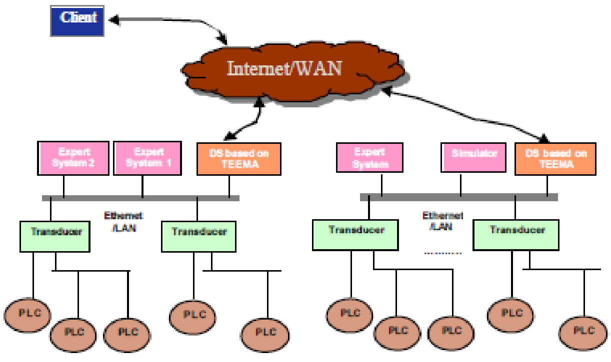


Fig.18 Agent-based SCADA architecture

The authors enabled software agents with a directory service to be applied to SCADA system architecture. The directory service is used mainly to provide the subsystem with the capability to locate where other subsystem services are located as shown in Fig.19.

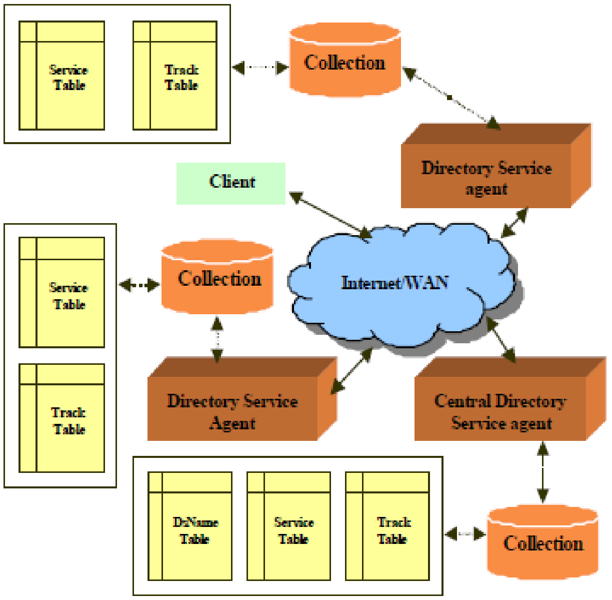


Fig.19 Prototype of the Directory Service System

A directory service is a shared information infrastructure for locating, managing, administering, and organizing common items and network resources. The used directory service is already provided by the used agent execution environment, which is called TEEMA (TRLabs Execution Environment for Software Agents) [43]. The usage of mobile agents over the Internet improves the versatility of the SCADA system. Not only each system is reliable and fault-tolerant in their site, but they are also able to access other remote systems if the need arises (local system partial failure). One important disadvantage of this system is its dependency on the availability of the Internet or other network connection.

4 DISCUSSION

Many other applications of the agent-based approach can be easily found in the industrial and MAS literatures. In this paper, we discussed a variety of the industrial applications of the agents and MAS paradigms to find the key conclusion, which we will use as a motivation to our future work for modeling and engineering of complex large-scale industrial environments. Industrial environments can be divided into two categories, continuous process industries, which include critical infrastructure utilities (i.e., electric power systems, water transportation, process control, etc), and discrete manufacturing industries, which include the common types of manufacturing systems [44]. Each of these two industrial environments has its own characteristics and requirements and the agent-based approach has been widely adopted to engineer both of them. This paper is concerned with the first category or the continuous process industries and specifically SCADA networks. The application of an agent-based infrastructure and architecture to a SCADA system is relatively straightforward in principle. However, some design issues need to be addressed at the very beginning of the characterization. First of all, it is necessary to identify the variation points of the SCADA system architecture, given an agent-based environment. There are three main features of an agent-based system that are advantageous for a large-scale SCADA system: modularity, communications, and mobility [42].

Now it is the time to ask "what is the common limitation in the provided agent-based solutions?" the answer of this question is that most of them adopt the agents and multi-agent paradigms with their traditional basic forms. In other words, the current practice of MAS in industrial environments tends to be limited to individual agents and static small face-to-face groups of agents that operate as closed systems, which comprise a small number of agents' types and a small number of agents in general [50]. This approach is not adequate to model and design complex, open, heterogeneous, and highly distributed industrial systems such as present and future complex large-scale SCADA networks, which are used to manage, supervise, and control the nations' critical infrastructures such as power grids (especially smart grids), water transportation, etc. Modern and future complex SCADA networks need to be flexible (i.e. able to adapt to the dynamic environments changes), robust (i.e. able to recover components failures), and adaptive (i.e. able to autonomously change their structures and organizations according to the context). For instance, consider the industrial environment shown in Fig.20. The figure presents a part of a complex smart grid. This type of systems is characterized by the following:

- Highly distribution
- Openness
- Heterogeneity
- Complexity
- Unpredictable and uncertain environment
- Large-scale nature

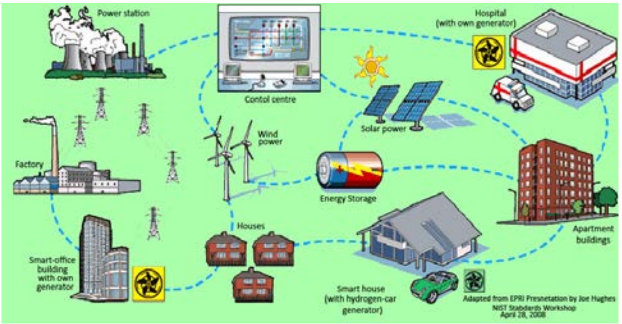


Fig.20 Part of a complex smart grid

When modeling this type of systems with the agent and MAS paradigms, the result will be a large number of agents interact with their environments and with each other as shown in Fig.21.

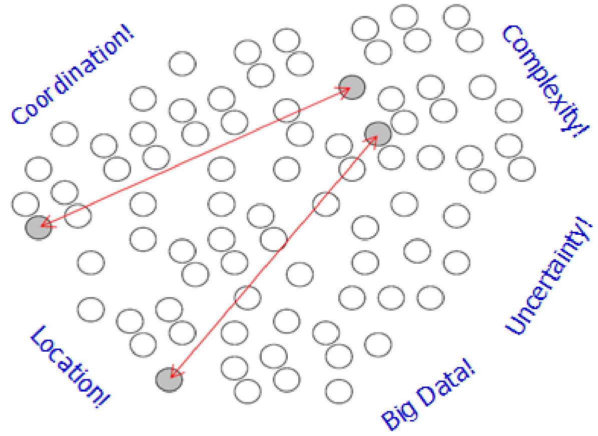


Fig.21 Agent-based modeling of smart grids

The problems that can appear are:

- Complexity: complex interactions and emergent behaviors.
- Coordination: How to align the behavior of different agents.
- Agent location: how can agents find each other to interact?
- Uncertainty: imperfect and unknown information can found within these systems.
- Big Data: These systems may include data sets with sizes beyond the ability of commonly used software tools to capture.

Although MAS are intrinsically autonomous, self-organized, and adaptive, but using them to develop complex, open, heterogeneous, and highly distributed systems such as modern and future SCADA networks will not be a trivial task. The reason of this is that MAS themselves need new software engineering methods, models, and techniques to get their full power. What complex MAS need are adaptive and dynamic organizational models that consider both of their aspects: their static/dynamic nature and their micro/macro levels. The MAS organizational models aim to enable the agents to dynamically reorganize to adapt environment dynamic and unpredictable changes. They give attention not only to the system micro level (individual agents) but also to the system macro level (the global system structure and behavior).

Although the HMS approach concerns the global system structure and behavior by adopting the concept of holons and hierarchies of holons (holarchies), but still it belongs to the hierarchical approaches. The HMS approach depends on the aggregation of holons, which means that one holon can be a part of another holon, thus the autonomy of a single holon is lower due to the fact that a holon is controlled and supervised by another higher level holon in system hierarchy. Further, the cooperation in HMS is always vertical and not horizontal. Luder et al. [45] stated that the aggregation is a key point in

HMS, where holons (agents) are structured in hierarchies; this is the appropriate solution to tackle the complexity of independent holons. This solution avoids complex communication and heavy network load in the systems, but takes a step back to centralized systems, causing a loss of flexibility and scalability of the system. Aggregation introduces new layers in the control pyramid and makes the logic of holons more complex. Even the communication is not getting simpler, because the holons have to communicate on different layers which results in process complexity.

Modern SCADA networks specially those which used to supervise and control critical infrastructures are highly distributed, complex, and large-scale [46]. They are not recursive in nature to be engineered by the holonic approach. To engineer this type of systems using the multi-agent paradigm, both of the global system structure and the local individuals should be concerned. Further, the predefined static organization used in the above explored systems is not flexible enough to allow them to adapt environment changes, and not robust enough to allow them to recover from failures. The research area which is concerned with this issue within MAS research is called MAS organization [47][48][49]. Therefore in the near future, a great attention should be given to proposing novel MAS organizational models similar to the holonic approach but carefully prepared to handle its limitations, for the design and development of large-scale real-life applications in general and complex industrial applications in specific.

5 CONCLUSIONS

The key conclusion of this paper is that using the agents and multi-agent software engineering paradigms with their traditional basic forms and concerning only the individual agents' behaviors without giving attention to the whole system structure and behavior is not adequate for modeling and developing complex large-scale industrial networks such as present and future SCADA systems. Also, the static and dynamic aspects of these systems must be simultaneously tackled through their whole development life-cycle for the sake of run-time stability and robustness. Our future research efforts aim to propose a matured dynamic MAS organizational model that tries to exploit the full power of MAS by giving the attention to the micro-macro and static-dynamic dimensions of MAS.

REFERENCES

[1] Luck, M. (1999). From definition to deployment: What next for agent-based systems?. The Knowledge Engineering Review, 14(02), 119-124.

[2] Willis, M. J. (1999). Proportional-integral-derivative control. Dept. of Chemical and Process Engineering University of Newcastle. Available on line at: http://educypedia.karadimov.info/library/PID.pdf

[3] Chen, C. T., & Hwang, C. (1990). Optimal on-off control for fed-batch fermentation processes. Industrial & engineering chemistry research, 29(9), 1869-1875.

[4] Weyns, D., Helleboogh, A., & Holvoet, T. (2009). How to get multi-agent systems accepted in industry?. International Journal of Agent-Oriented Software Engineering, 3(4), 383-390.

[5] Metzger, M., & Polakow, G. (2011). A survey on applications of agent technology in industrial process control. Industrial Informatics, IEEE Transactions on, 7(4), 570-581.

[6] Pereira, N. Rodrigues, and P. Leitão," Deployment of Multi-agent Systems for Industrial Applications", 17th IEEE International Conference on Emerging Technologies and Factory Automation, 2012, pp. 1-8.

[7] F. Bergenti and E. Vargiu. "Multi-Agent Systems in the Industry. Three Notable Cases in Italy." WOA. 2010.

[8] F. Sayda, Multi-agent systems for industrial applications: design, development, and challenges. INTECH Open Access Publisher, 2011.

[9] M. Pěchouček and V. Mařík. "Industrial deployment of multi-agent technologies: review and selected case studies." Autonomous Agents and Multi-Agent Systems 17.3 (2008): 397-431.

[10] Maturana, F., Staron, R., Tichy, P., Vrba, P., Rischar, C., Mařik, V., & Hall, K. (2006). An Agent-OS for Autonomous Cooperative Systems. In Information Technology For Balanced Manufacturing Systems (pp. 29-36). Springer US.

[11] Maturana, F., Staron, R., Hall, K., Tichý, P., Vrba, P., & Mařík, V. (2008). Agent virtual machine for automation controllers: Example application: Shipboard automation. Robotics and Computer-Integrated Manufacturing, 24(5), 616-624.

[12] Monostori, L., Váncza, J., & Kumara, S. R. (2006). Agent-based systems for manufacturing. CIRP Annals-Manufacturing Technology, 55(2), 697-720.

[13] Parunak, H. V. D. (1998). Practical and industrial applications of agent-based systems. Environmental Research Institute of Michigan (ERIM).

[14] Shen, W., & Norrie, D. H. (1999). Agent-based systems for intelligent manufacturing: a state-of-the-art survey. Knowledge and information systems, 1(2), 129-156.

[15] Diep, D., Massotte, P., & Meimouni, A. (2003, August). A distributed manufacturing execution system implemented with agents: the PABADIS model. In Industrial Informatics, 2003. INDIN 2003. Proceedings. IEEE International Conference on (pp. 301-306). IEEE.

[16] Botti, V., & Boggino, A. G. (2008). ANEMONA: A multi-agent methodology for Holonic Manufacturing Systems. Springer Science & Business Media.

[17] Hayashi, H. (1993). The IMS International Collaborative Program. In Proceedings of 24th ISIR, Japan Industrial Robot Association. (Earlier description of the Intelligent Manufacturing Systems international research program. The HMS consortium is one of those operating under this program.) (http://www.ims.org/)

[18] Koestler, Arthur (1967). The Ghost in the Machine (1990 reprint ed.). Penguin Group. ISBN 0-14- 019192-5

[19] Van Leeuwen, E.H. and Norrie, D.H. (1997). Intelligent manufacturing: holons and holarchies. Manufacturing Engineer, 76(2), 86-88.

[20] W. Bolton, "Programmable Logic Controllers," 5th ed., Newnes, 2009 ISBN 978-1-85617-751-1, Chapter 1.

[21] Leitao, P. and Restivo, F. (2006) 'ADACOR: a holonic architecture for agile and adaptive manufacturing control', Computers in Industry, Vol. 57, No. 2, pp.121–130, Elsevier.

[22] Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., & Peeters, P. (1998). Reference architecture for holonic manufacturing systems: PROSA.Computers in industry, 37(3), 255-274.

[23] Giret, A., & Botti, V. (2004). Holons and agents. Journal of Intelligent Manufacturing, 15(5), 645-659.

[24] Bratoukhine, A., Penya, Y. K., & Sauter, T. (2002). Intelligent Software Agents in Plant Automation. In in Proceedings of the 2002 NAISO Symposium on Engineering of Intelligent Systems (EIS), pg 77.

[25] Weedy BM, Cory BJ (1998) Electric Power Systems. Wiley, Chichester

[26] Preiss O, Wegmann A (2001) Towards a composition model problem based

on IEC61850. Proceedings of the 4th ICSE Workshop on Component-based Software Engineering, Toronto, Canada.

[27] McArthur, Stephen DJ, et al. (2007), "Multi-agent systems for power engineering applications—Part I: concepts, approaches, and technical challenges." Power Systems, IEEE Transactions on 22.4: 1743-1752.

[28] Kok, J. K., C. J. Warmer, and I. G. Kamphuis.(2005) "PowerMatcher: multiagent control in the electricity infrastructure." Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems. ACM.

[29] Davidson, E. M., McArthur, S. D., McDonald, J. R., Cumming, T., & Watt, I. (2006). Applying multi-agent system technology in practice: automated management and analysis of SCADA and digital fault recorder data. Power Systems, IEEE Transactions on, 21(2), 559-567.

[30] McArthur, S. D., Davidson, E. M., Hossack, J. A., & McDonald, J. R. (2004, January). Automating power system fault diagnosis through multi-agent system technology. In System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on (pp. 8-pp). IEEE.

[31] F. Bellifemine, A. Poggi, G. Rimassi. (1999), "JADE: A FIPA-Compliant Agent Framework." Proceedings of the Practical Applications of Intelligent Agents and Multi-Agents, April 1999; 97-108.

[32] Jennings, N.R., Corera, J.M. and Laresgoiti, I. (1995) 'Developing industrial multi-agent systems', in Proceedings of the First International Conference on Multi-agent Systems (ICMAS-95), pp.423–430.

[33] T. Wittig, N. R. Jennings, and E. M. Mandan, "ARCHON:—a framework for intelligent co-operations," IEE-BCS J. Intell. Syst. Eng., vol. 3, no. 3, pp. 168–179, 1994.

[34] P. D. Anh and T. D. Chau, Component-based Design for SCADA Architecture, International Journal of Control, Automation, and Systems (IJCAS), vol. 8, no. 5, 2010, pp.1141-1147.

[35] Abbas, H.A. and Mohamed, A.M. (2011) 'Review in the design of web based SCADA systems based on OPC DA protocol', International Journal of Computer Networks, February, Vol. 2, No. 6, pp.266–277, Malaysia.

[36] Mohamed, A.M. and Abbas, H.A. (2011) 'Efficient web based monitoring and control system', in Proceedings of the Seventh International Conference on Autonomic and Autonomous Systems, ICAS 2011, May 22–27, 2011, Venice, Italy.

[37] E. Diaconescu and C. Spirleanu, Communication solution for industrial control applications with multi-agents using OPC servers, In Applied and Theoretical Electricity (ICATE), 2012 International Conference on, 2012, pp. 1-6.

[38] OPC Foundation, "OPC DA 3.0 Specification [DB/OL]", Mar.4, 2010

[39] W. Farai Rupare, L. Nyanga, A. van der Merwe, S. Mhlanga, and S. Matope, "Design Of An Automated Grinding Media Charging System For Ball Mills", SAIIE25 Proceedings, 9th – 11th of July 2013, ref no. 619, pp. 507-518, Stellenbosch, South Africa.

[40] Jacke, B., Sauer, O., & Sutschet, G. (2006). ProVis. Agent: An Agent-Based Production Monitoring And Control System. In Proc. of the Workshop on Software-Agents in Information Systems and Industrial Applications (SAISIA), Fraunhofer IRB Verlag (pp. 75-82).

[41] Mangina, E. E. (2005). Intelligent Agent-Based Monitoring Platform for Applications in Engineering. IJCSA, 2(1), 38-48.

[42] Prayurachatuporn, S., & Benedicenti, L. (2001). Increasing the reliability of control systems with agent technology. ACM SIGAPP Applied Computing Review, 9(2), 6-12.

[43] Martens, R., W. Hu, et al. 2001. TEEMA: TRLabs Execution Environment for Mobile Agents. TRLabs Regina, Regina, Saskatchewan, Canada (Internal Report).

[44] Buse, D.P. and Wu, Q.H. (2007) IP Network-based Multi-agent Systems for Industrial Automation: Information Management, Condition Monitoring and Control of Power Systems, Springer.

[45] Arndt Lüder, Axel Klostermeyer, Jörn Peschke, Alexei Bratoukhine, and Thilo Sauter (2005), "Distributed Automation: PABADIS versus HMS", IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 1, NO. 1, FEBRUARY 2005

[46] Abbas, H.A. (2014), Future SCADA challenges and the promising solution: the agent-based SCADA", Int. J. Critical Infrastructures, Vol. 10, Nos. 3/4, pp.307–333.

[47] Abbas, H. A.; Shaheen, I.S; Amin, M. H. (2015); Organization of Multi-Agent Systems: An Overview, International Journal of Intelligent Information Systems. Vol. 4, No. 3, 2015, pp. 46-57. doi: 10.11648/j.ijiis.20150403.11.

[48] Abbas, H. A. (2014). Exploiting the Overlapping of Higher Order: Entities within Multi-Agent Systems. International Journal of Agent Technologies and Systems (IJATS), 6(3), 32-57.

[49] Abbas, H. A. (2015), Realizing the NOSHAPE MAS Organizational Model: An Operational View, International Journal of Agent Technologies and Systems (IJATS), Volume 7, Issue 2, Pages 75-104, IGI Global.

[50] Odell, J. J., Parunak, H. V. D., & Fleischer, M. (2003). The role of roles in designing effective agent organizations. In Software Engineering for Large-Scale Multi-Agent Systems (pp. 27–38). Springer Berlin Heidelberg. doi:10.1007/3-540-35828-5_2.