VTT Technical Research Centre of Finland

Proactive computing in process monitoring

Pakonen, Antti; Pirttioja, Teppo; Seilonen, Ilkka; Tommila, Teemu

Published in:
Proceedings

DOI:
10.1109/ETFA.2006.355408

Published: 01/01/2006

Document Version
Peer reviewed version

Proactive Computing in Process Monitoring: Information Agents for Operator Support

Antti Pakonen[1], Teppo Pirttioja[2], Ilkka Seilonen[3], Teemu Tommila[1]

[1]VTT Technical Research Centre of Finland, e-mail: antti.pakonen@vtt.fi
[2]Automation Technology Laboratory, Helsinki University of Technology, Finland, e-mail: teppo.pirttioja@tkk.fi
[3]Information and Computer Systems in Automation, Helsinki University of Technology, e-mail: ilkka.seilonen@tkk.fi

Abstract-**While automation systems can track thousands of measurements it is still up to human process operators to determine the operational situation of the controlled process, particularly in abnormal situations. To fully exploit the computing power of embedded processors and to release humans from simple data harvesting activities, the concept of proactive computing tries to exploit the strengths of both man and machine. Proactive features can be implemented using intelligent agent technology, enabling humans to move from simple interaction with computers into supervisory tasks. Autonomous information agents can handle massive amounts of heterogeneous data. They perform tedious tasks of information retrieving, combining and monitoring on the behalf of their users. This paper presents a multi-agent-based architecture for process automation, which aims to support process operators in their monitoring activities. The approach is tested with a scenario inspired by a real-world industrial challenge.**

I. Introduction

With the increasing use of instrumentation and control systems, a smaller number of process operators is in charge of ever larger portions of manufacturing processes. While the detection of typical disturbances can fairly well be automated, it is still mostly up the human users to spot unexpected abnormal situations, and to judge the operational state and performance of process equipment.

Automation systems can track tens of thousands of measurements at a rate superior to humans. Although human decision making is certainly needed on the higher level, we cannot expect that process operators use all the sensor data in their work. User interfaces should be designed to utilise the strengths of both humans and computers in order to compensate for their weaknesses.

This paper presents a multi-agent-based approach to process automation intended to support the situation awareness and decision-making of process operators as well as maintenance personnel or shift supervisors. A group of intelligent information agents can flexibly combine heterogeneous information about the process situation from many different computer systems typically used in process plants. An operator can easily configure intelligent monitoring and information retrieval services. The agent-based approach outlined in the paper would enable easy incorporation of new functionality on top of existing control and information systems.

The paper is structured as follows: Section II describes some challenges in present-day process monitoring. Section III shortly introduces the related concepts of proactive computing and indirect management. Section IV introduces some ideas of agent-based computing. An agent-based architecture for process automation is outlined in Section V. Section VI presents a demonstration scenario motivated by a real-world process monitoring problem, and the conclusions are in Section VII.

II. Operational Monitoring of Process Plants

A. Human-Computer Interaction

Monitoring of a process plant is usually the responsibility of a process operator. The objective is to detect disturbances as early as possible in order to minimise losses in production and ensure efficient functioning of the factory process [1]. The system interface used by an operator is typically a display representing the structure of the controlled process, with signal values or trend graphs to show the state of key process variables. Traditionally, monitoring is automated through predefined alarms that are all but flexible and often result in alarm floods during unexpected situations. The automatic recognition of abnormal situations or the operational state of the process as a whole is still very hard [2].

As the amount of personnel in factories is on a steady decline, the sphere of responsibilities of an individual operator is broadening [2]. In comparison, humans are more intelligent and flexible than computers, but less reliable and with limited capacity to process information [1]. Since a process can contain thousands of measurements, the operator cannot be expected to control the operational state of the process by browsing through them all. The user interface should be designed keeping this in mind.

B. Distribution of Process Information

Standardisation of automation has thus far mostly dealt with the lowest levels. We have standards for e.g. field bus communication, but on higher levels the semantics of different systems are all but compatible [2]. To worsen the situation, the life-cycle of applications in process automation is fairly long, and open standards and solutions are not favoured by many major vendors. As a result, process information is located on a large number of systems and databases with different interfaces, semantics, syntaxes, and data formats.

When a process operator sees an anomaly on a measurement trend, he may easily conclude that something has possibly gone wrong. However, the reason behind the anomaly can be the replacement or calibration of a sensor device by a maintenance person, instead of a malfunction. The maintenance report is saved to a maintenance database, but the operator may not realise or may simply forget to look for that information. As this example highlights, all necessary information does not come from the measurements. The users of process automation should have efficient methods to access different systems and databases, corporate intranets, the Internet, etc. in order to be able to correctly interpret the situation [2]. Many process operators frankly admit to stumbling to simple problems such as not remembering the user passwords for a given system interface.

The leading principle in the process automation user interface design should be *right information at the right time in the right form* [2]. It should not be up to the user to know what the appropriate sources for relevant information are, or how the information in the source can be retrieved.

III. Proactive Computing and Information Processing

Automation systems, or embedded computer systems in general, can observe and control their physical environment with real time responsiveness. The ever increasing amount of embedded processors and sensors directly access tremendous amounts of information about the real world. In traditional human-computer interaction (writing a document at the office), information is mostly meditated by humans. When facing the challenges of the real world, however, time constants diminish and the amount of data increases dramatically. Accordingly, the role of humans must shift from interaction to supervision. In *proactive computing* [3], embedded computers monitor and shape their environment without direct human involvement. Operation is neither human-controlled nor completely automatic, but human-supervised. For as much as possible, human users are only involved in policy-making roles and in providing guidance in critical decisions [4].

The concept of proactivity is frequently used when describing agent-based systems. In user interface research, the style of interaction where humans manage and supervise intelligent, proactive computer agents that perform and even initiate tasks on their behalf has been referred to as *indirect management* [5]. The user and his agent assistant are engaged in a cooperative process in which both of them can initiate communication and perform tasks. The agent assists the user by hiding the complexity of difficult tasks, by helping different users collaborate, and by monitoring for events [6]. In a vast network of information, agents can be used for information searching and filtering.

IV. Information Agents

Agent technology is a novel computing paradigm aimed at managing complex, distributed systems [7]. An agent is usually characterised as an autonomous and flexible entity capable of proactive, goal-based action on its environment. As a society, agents in multi-agent systems (MAS) can negotiate and bargain about their goals in order to adapt to dynamic, complex environments.

A. Information Agents

Since the breakthrough of Internet, it has been envisioned that one day, intelligent agents would be able to retrieve information from all over the Web on behalf of us. The heterogeneity and sheer amount of information sources could be managed with tireless virtual actors composed of the fabrics of the Web itself.

For the lack of established terminology and definitions, different researchers talk of "intelligent software agents" [8] or "information agents" [9] while essentially meaning the same thing. In our research, we have found the term "information agent" useful in emphasising the distinction from activities directly linked to real time control. Obtaining a definition from [9], *information agents* are intelligent agents that have access to heterogeneous and geographically distributed information sources, and witch *proactively* acquire, mediate, and maintain relevant information *on behalf of users* or other agents.

Information agent systems tend to be realised as multi-agent systems (MAS) in order to deal with the dynamic changes in the information source landscape [8]. The multi-agent approach is well suited to handle the heterogeneity of the information sources, and the related information needs of multiple users. Multi-agent systems offer concurrency, modularity and robustness, and facilitate the efficient distribution of data processing to the sites where the data resides [10].

B. Belief-Desire-Interaction Architecture

The Belief-Desire-Intention (BDI) Architecture [11] is a popular agent model that defines a rational agent with mental attitudes; *beliefs* that represent information about the state of the agent's environment, *desires* that denote the objectives adopted by the agents, and *intentions* that represent the currently chosen course of action. The implementation of a BDI system is often based on *plans*, abstract specifications of how to achieve certain desires.

In information gathering, multi-agent based distributed query planning enables dynamic processing of complex queries and provides concurrency when the search space is very large [10]. Information agent plans can be constructed to facilitate a range of access methods, communication protocols and data formats.

C. Ontologies

For successful co-operation in multi-agent systems, an agreement on the *semantics* of the information exchanged between agents is needed. A common *ontology* defines the meanings of the concepts of a domain in a machine-processable format and facilitates the processing of heterogeneous information. A formal conceptual model supports automatic reasoning and combining of information in different syntaxes and on different levels of abstraction.

Ontologies have been developed in the semantic web research area to support searching and interpretation of the massive amounts of heterogeneous information in the Internet. The current W3C standard for ontological modelling is the Web Ontology Language (OWL) [12]. Because of the tools readily available for OWL, it has gained popularity in MAS research, also in the application domain of automation [13].

V. MULTI-AGENT SYSTEM ARCHITECTURE

We have developed a multi-agent architecture for process automation, aimed at assisting human users in operational monitoring and maintenance tasks. The architecture is influenced by our earlier work on process control agents [14] and by the general research on information agent technology [9][15].

The agents reside on an agent platform on top of a control system (Fig. 1). They extract information from the underlying control hardware (e.g. PLC, DCS, PC) or directly from intelligent sensor devices via a fieldbus. They also search and fuse information from various different IT systems, such as:

- Manufacturing Execution Systems (MES)
- Supervisory Control and Data Acquisition (SCADA) software
- Measurement and alarm history databases
- Maintenance databases
- Electronic diaries
- Corporation intranet (with e.g. design documentation)

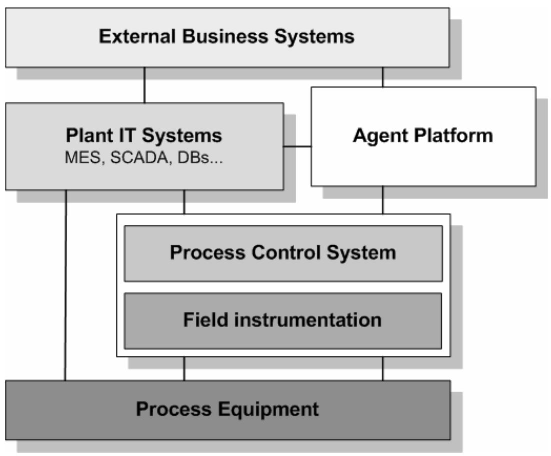

Fig. 1. The agent platform resides on top of the physical control system.

The agents can also connect to external business systems (ERP, SCM), or even to the Internet. All data extracted by the agents is given a semantic mapping to the ontology used by the agent society to facilitate information processing.

In many ways, the agent platform works in parallel with SCADA systems. They exchange information, and the functions provided by agents are similar, yet more developed than those of the current user interfaces. Our add-on approach has enabled us to experiment with present-day agent implementation tools without interfering with the real time control tasks.

A. Different Roles in the Agent Society

The agents in our multi-agent society operate in different roles. The roles provide a hierarchy for agent organisation and define the responsibilities, goals and behaviour of individual agents. The clear definition of goals or objectives is the basis for the goal-driven action planning of a single agent.

We propose a multi-agent architecture of agents in five different roles (Fig. 2). The distribution of roles is motivated by the range of responsibilities and tasks that both different automation system components and different system users have.

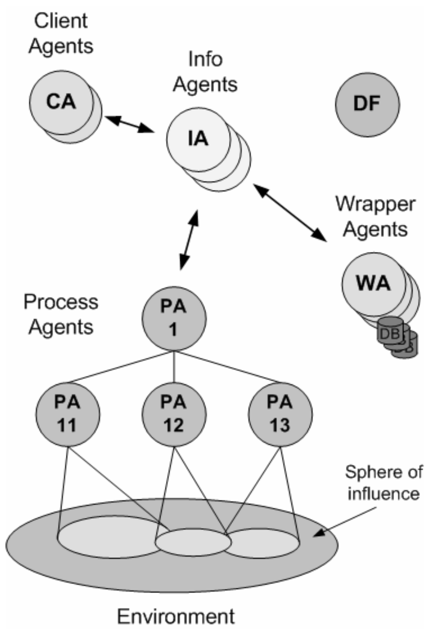

Fig. 2. The agent society consists of agents in five different roles.

The *Directory Facilitator* (DF, standardised by FIPA [16]) is a yellow pages directory service used by the agent society to advertise and locate services provided by other agents.

Wrapper Agents provide transparent access to different legacy data sources, such as measurement or maintenance history databases. Their objectives are 1) to maintain a data connection, 2) to translate data from different formats to information understood by other agents, and 3) monitor for changes or updates in the information stored in the data source.

The process equipment is represented by a group of *Process Agents*. Each Process Agent corresponds to a physical section of the process plant. The hierarchical composition extends from low-level agents representing single devices (pumps or valves) to high-level agents responsible for entire processes. The objectives of the Process agents are 1) to collect and refine information about the operational state of the equipment, 2) to discover deficiencies in the performance of the devices and 3) to monitor for process phenomena of interest.

Client Agents provide different interfaces for human users in different roles (e.g. process operator, maintenance person, or shift supervisor). Their objective is 1) to support the tasks of the human user by providing information retrieval services and 2) to maintain a conversation with the human users in accordance with the current agent task. A separate agent interface is not needed, if new functionality can be incorporated into the existing user interface.

The above-mentioned agents have permanent task assignments, but the lifespan of an *Info Agent* is that of the given information retrieval or monitoring task. They are activated when necessary to perform tasks that require combining information from different sources. Their objectives are 1) to decompose information requests to a set of data queries to the appropriate sources, 2) to filter and format the collected data, and 3) to monitor for changes in the desired information.

B. Agent Structure

The internal structure of an agent (Fig. 3) is based on the BDI agent model. The BDI *planner module* controls the use of dedicated information processing modules. Depending on adopted goals or the state of the environment (*beliefs*), suitable plans are selected from the *plan library*. Some of the goals are permanent (e.g. observing the operational state of the physical process) while some goals are adopted in order to perform given tasks. Task requests may come from human users or other agents. Agent communication takes place on the level of intentions (goals) - it is up to the agent to determine how the desired high-level goal is reached.

Different information processing modules are used to access the data source (over e.g. OPC or ODBC), to process the obtained data (using appropriate symbolic or mathematic reasoning methods), and to communicate between agents. Different data source query plans constantly update the agent's beliefs, allowing the retrieved information to be readily accessed by the various information processing plans.

The use of goal-based planning for process monitoring is more thoroughly examined in [17].

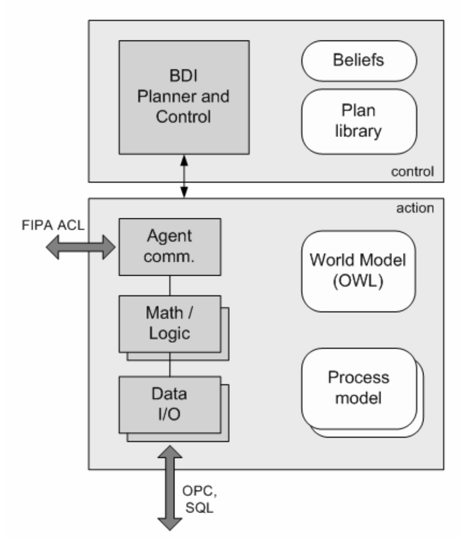

Fig. 3. An agent consists of a BDI planner unit, and several information processing units.

C. Related Research

Much of the research on agent-based automation has focused on distributed control of discrete manufacturing (e.g. Holonic Manufacturing Systems [18]). Nevertheless, multi-agent systems have also been proposed for the handling of process information. Applications include process monitoring, diagnostics, and alarm handling [19][20][21]. It is also worth noting that a function similar to the one demonstrated in the next chapter has been proposed in [22], although the system implementation is different.

VI. Demonstration

A. Demonstration Setting

The feasibility of the agent-based architecture has already been tested in demonstration scenarios [17]. We have implemented our agent architecture using a publicly available, open-source Java agent development tool called Jadex [23]. Jadex is an extension of the popular agent platform JADE, with agent structure based on the BDI-model.

The background for the scenarios has been an actual industrial process, a pulp bleaching plant of a paper mill. In the hostile environment of the plant (e.g. vibration and exposure to corrosive chemicals) sensitive measurement devices can easily malfunction.

The problems are highlighted if the fault in a sensor is developing slowly. For example, after the actual bleaching reaction under alcaline conditions, the pulp is neutralised by using sulphur dioxide. The dosage of the sulphur dioxide is based on a sensitive pH measurement. If the pH sensor fails and the measurement slowly drifts to inaccurate values, the control loop can compensate the drift by increasing the dosage. On the subprocess level, everything seems to be ok.

The problem in the dosage can only be detected by comparing it to the situation as a whole.

This fairly typical phenomenon is well known by plant personnel, and it is usually noticed. However, if the operator is busy in doing other things e.g. during startup, the fault can go on unnoticed for several hours, resulting in masses of spoiled pulp. To fix things, the operator should be able to easily configure and manage a monitoring service capable of combining necessary measurements. Instead of fixed alarm limits, we need a simple assistant to *temporarily* help us complete given tasks (e.g. a flawless startup).

B. Using the Monitoring Agent

In our case, the correct dosage of the sulphur dioxide can be estimated on the basis of the dosage of bleaching chemicals before the actual bleaching reaction. However, these variables are part of another subprocess that is physically located elsewhere. To compare the values, the user may have to continuously browse through multiple displays. To fix this, the agents offer a monitoring service, where the monitored variables are added in a drag-and-drop fashion from the standard process view. The user then defines simple constraints that define the normal operation of the subprocess of interest.

After the constraints for correct operation have been defined, it is up to the agents to locate the sources of needed information, and to monitor the process measurements. The monitoring agent now acts as an assistant of the user, taking care of the tedious, simple tasks involved. Instead of continuously browsing through all the data, the operator now supervises an agent that does it for him.

C. Agent Interaction

The demonstration involves a Client Agent providing the user interface, an Info Agent to construct and carry out the monitoring task, a couple of Process Agents to provide the measurement data, and the Directory Facilitator (Fig. 4). The conversations between the agents follow conversation protocols for e.g. information *subscription* instead of simple query-response polling.

Upon receiving the monitoring task the Info Agent sets about to perform the actions needed to fulfil the user-defined objective. Specialised BDI *plan* files define actions for retrieving and processing of information. These actions are appropriately combined to achieve the needed steps to perform the monitoring task;

1) Parse the query to discover the needed measurements.

2) Find the data sources for the measurements needed. The appropriate source agents for each measurement can be found by querying the Directory Facilitator.

3) Divide the query into subqueries for each data source. Using e.g. Constraint Satisfaction Problem (CSP) [24] tools, the Info Agent can divide the original query containing all the conditions into new subsets of conditions that apply to measurements held by an individual Process Agent.

4) Request the Process Agents involved to notify when the conditions are breached. Until one of the Process Agents notifies a violation in the conditions, no communication between the agents takes place. The polling of data has been allocated as close to the information source as possible, reducing the amount of necessary communication. If and when the conditions are violated, the Info Agent is now able to directly inform which of the conditions went off and supply the user with the measurement data involved.

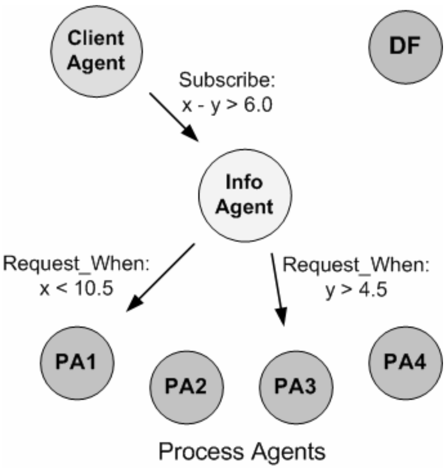

Fig. 4. The Info Agent carries out the monitoring task by dividing the set of monitored measurements (labelled x and y for simplicity) to appropriate Process Agents. Conversations follow protocols defined by FIPA.

The agent system provides a reconfigurable platform for easy development and maintenance of process automation monitoring services. The BDI plans define a generic monitoring functionality that can be mapped into a completely different process plant with different data sources and interfaces only by redefining the data connection configuration files (written in XML in our implementation). To incorporate new functionality into the monitoring activities, only the BDI plan files for monitoring (XML in the Jadex platform) must be altered. Different subplans for e.g. information retrieval are designed to be reusable in different task contexts - much of the communication between plans takes place through the agent's *beliefs*.

The Constraint Satisfaction Problem (CSP) tool that was used in the demonstration is only one of the possible tools to process information or to decompose tasks. Using BDI planning, appropriate methods can automatically be selected to produce the desired output from the source data. For the user, the sources of information as well as the methods used for information processing are transparent. Using proactive plan-based task decomposition, the agents can flexibly perform any kind of monitoring task with any kind possible of value constraints.

D. Further Issues

The user interface developed for the demonstration is still tentative. Although the measurements can be added in a drag-and-drop fashion, the way to specify the monitoring constraints between them needs further thought.

It is also intended, that the user could as easily define conditions on the duration of the monitoring task, and criteria for process situations (e.g. maintenance or shutdown) where the monitoring rules should be ignored. While this demonstration scenario only exploited measurement data, further test scenarios will incorporate more diverse information from different sources.

VII. CONCLUSIONS

In this paper, we presented a proactive, multi-agent-based approach for operational monitoring of industrial processes. We demonstrated how information agents can support and ease the work of human users in a case scenario motivated by a real world problem. Our research shows that agent-based processing of process plant information can facilitate efficient process monitoring and also free human users from tedious information retrieval activities and emphasise their role as decision makers.

The functionality demonstrated here could of course also be achieved using other technologies. However, we claim that especially the development and maintenance of monitoring applications would significantly be rationalised by using agent and BDI technology. An interesting alternative to MAS would be Intelligent Web Service technology, which attempts to tackle same issues as information agents from another viewpoint. What the actual enabling technologies for the Web of the future will be remains to be seen. We believe that future process automation systems will possess at least some agent-like characteristics, even if multi-agent systems as such will not be widely adopted.

Some development steps are needed before agent-based systems can be applied to industrial production plants. We still lack industrial-strength software tools for agent development. The methods for agent-oriented software engineering (AOSE) as well as the methods for semantic web application development are still largely under construction.

In the future, we aim to test our multi-agent architecture in further real-world scenarios. We will attempt to fuse more diverse information from the different IT systems of the process plant to facilitate the decision-making of not only the process operators, but also maintenance personnel and e.g. shift management. We believe that the agent-based approach would facilitate more efficient cooperation between human users in different roles.

Before proactive computing can be fully utilised in process automation, further research into the user interfacing of the agent system is also needed. The agent-based approach forces us to thoroughly reconsider how human users and process automation systems should interact.

REFERENCES

[1] H. Paunonen, *Roles of informating process control systems*, Dissertation, Tampere, Finland: Tampere University of Technology, 1997.

[2] O. Ventä, "Research view of Finnish automation industry," *Industrial Systems Review 2005*, pp. 48-55, VTT Industrial Systems, 2005.

[3] D. Tennenhouse, "Proactive computing," *Communications of the ACM*, vol. 43, pp.43-50, May 2000.

[4] R. Want, T. Pering, D. Tennenhouse, "Comparing autonomic and proactive computing," *IBM Systems Journal*, vol. 41, pp. 129-135, 2003.

[5] A. Kay, "User interface: a personal view," B. Laurel (editor), *The Art of Human-Computer Interface Design"*, Addison-Wesley, 1990.

[6] P. Maes, "Agents that reduce work and information overload," *Communications of the ACM*, vol. 37, pp. 30-40, July 1994.

[7] N.R. Jennings, "On agent-based software engineering," *Artificial Intelligence,* vol. 117, pp. 277-296, 2000.

[8] K. Sycara and D. Zeng, "Multi-agent integration of information gathering and decision support," *12th European Conference on Artificial Intelligence*, pp. 549-556, 1996.

[9] M. Klusch, "Information agent technology for the Internet: a survey," *Data & Knowledge Engineering,* vol. 36, pp. 61-47, 2001.

[10] T. Oates, M.V. Nagendra Prasad and V.R. Lesser, "Cooperative information-gathering: a distributed problem-solving approach," *IEE Proceedings on Software Engineering, Special Issue on Agent-based Systems*, vol. 144, pp. 72 - 88, 1997.

[11] A.O. Rao and M.P. Georgeff, "BDI agents: from theory to practice," Technical Note 56, Australian Artificial Intelligence Institute, Melbourne, Australia, 1995.

[12] OWL Web Ontology Language, W3C Recommendation, http://www.w3c.org/TR/olw-ref/, 2004.

[13] M. Obitko, V. Marik, "Adding OWL Semantics to ontologies used in multi-agent systems for manufacturing," *First International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2003*, Prague, Czech Republic, September 1-3, 2003.

[14] I. Seilonen, T. Pirttioja, P. Appelqvist, A. Halme and K. Koskinen, "An approach to process automation based on cooperating subprocess agents," *First International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2003*, Prague, Czech Republic, September 1-3, 2003.

[15] N. Nodine, A.H.H. Ngu, A. Cassandra, W.H. Bohrer, "Scalable semantic brokering over dynamic heterogeneous data Sources in InfoSleuth," *IEE Transactions on Knowledge and Data Engineering*, vol 15, no. 5, 2003.

[16] The Foundation for Intelligent Physical Agents, FIPA, http://www.fipa.opg/.

[17] T. Pirttioja, A. Pakonen, I. Seilonen, A. Halme and K. Koskinen, "Multi-agent based information access services for condition monitoring in process automation," *3rd International IEEE Conference on Industrial Informatics, INDIN 2005*, Perth, Australia, August 10-12, 2005.

[18] S. M. Deen (editor), *Agent-Based Manufacturing, Advances in the Holonic Approach*, Springer-Verlag Heidelberg 2003.

[19] J. Corera., I. Laresgoiti., N. R. Jennings, "Using Archon, part 2: electricity transportation management," *IEEE Expert*, vol. 11, no. 6, pp. 71-79, 1996.

[20] L. Bunch, M. Breedy, J. M. Bradshaw, M. Carvalho, N. Suri, A. Uszok, J. Hansen, M. Pechoucek, V. Marik, "Software agents for process monitoring and notification," *2004 ACM Symposium on Applied Computing*, pp. 94-99, ACM 2004.

[21] H. Wörn, T. Längle, M. Albert, "Multi-agent architecture for monitoring and diagnosing complex systems," *The 4th International Workshop on Computer Science and Information Technologies*, 2002.

[22] T. Koskinen, M. Nieminen, H. Paunonen and J. Oksanen, "The framework for indirect management features of process control user interfaces," *10th International Conference in Human - Computer Interaction*, Crete, Greece, June 22-27, 2003.

[23] Jadex BDI Agent System, University of Hamburg, http://vsis-www.informatik.uni-hamburg.de/projects/jadex/

[24] E. P. K. Tsang, *Foundations of Constraint Satisfaction,* Academic Press, London and San Diego, 1993.