VTT Technical Research Centre of Finland

Multi-agent based information access services for condition monitoring in process automation

Pirttioja, Teppo; Pakonen, Antti; Seilonen, Ilkka; Halme, Aarne; Koskinen, Kari

Published in:
3rd IEEE International Conference on Industrial Informatics, INDIN '05

DOI:
10.1109/INDIN.2005.1560383

Published: 05/08/2005

Document Version
Early version, also known as pre-print

Multi-Agent Based Information Access Services for Condition Monitoring in Process Automation

Teppo Pirttioja[1], Antti Pakonen[2], Ilkka Seilonen[3], Aarne Halme[1], Kari Koskinen[3]

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

***Abstract*—This paper studies issues concerning the application of goal-oriented information agents to condition monitoring tasks in process automation. In the presented approach, an agent-based information access layer is operating as a mediator for various types of process-related information. The agent system may be seen as an active partner for human users in this information intensive application domain. In the future, more and more automation related information will be available in electronic form. Physical sensors provide not only measurements but also reliability information, and numerous configuration details are available in electronic form. The challenge is to integrate as much of the available information as possible, and to automate the processing of this information to enable easy decision-making for end users.**

***Index Terms*—process automation, condition monitor, agent, ontology, services.**

I. Introduction

Automation has become an information intensive application domain in which more and more information is available in various electronic forms. Although we have access to all the information needed for concluding the condition of the physical process equipment, systems designers have no effective methods for achieving this. There is a demand for system architectures and engineering tools that could be used as a basis for building advanced condition monitoring services. These services would assist human users in evaluating the performance of an increasingly complex process setup.

This paper is motivated by the fact that information agent based solutions have been successfully applied to domains that have a number of problems similar to condition monitoring in automation. Information agents perform well in an environment that is distributed and dynamically changing by nature.

This paper is outlined as follows: Section II will discuss the state-of-the-art in automation and information agent research. Section III presents our agent-based architecture and Section IV focuses on general operation principles. A demonstration based on our approach is presented in Section V, and final conclusions are in Section VI.

II. Automation and Information Agents

The amount of available measurement data in process automation systems is increasing rapidly. With the increase in instrumentation and the increase of intelligence in the field devices, we have more than enough of process data. However, the operational state of the process is still difficult to determine [22][23]. It is hard to recognize abnormal situations, or furthermore trace back their causes.

It is also apparent, that all process information is not derived directly from the measurements. We would like to utilize information available in e.g. design documents, corporate intranets, the Internet, and particularly different process models and simulators in process monitoring [23]. In the complex and dynamic environment of process automation, combining these systems presents us with two key challenges. First, although numerous interfacing technologies like OPC or OBDC practically provide access to all information, pieces of information cannot be fused together. Because of the many standards in the field, information is represented in different syntaxes, different semantics, and on different levels of abstraction. Second, process models and simulators cannot be utilized in monitoring, be-cause they only apply to a certain operational state, and the actual state of the process is not known. An evident example of the situation is the inadequate processing of alarms; in the event of a severe fault the user is flooded with alarms, but the actual cause of the fault often remains unknown.

The traditional approach to operational state management has been the use of the so-called expert systems. As they are typically based on a centralized rule base, adapting to changes in the system or environment calls for a huge amount of reconfiguration work [22]. Also, the most advanced systems for alarm flood filtering available rely on simple rules defining the criticality of each individual alarm.

In other domains, solutions based on intelligent and autonomous information agents have been adopted to enable uniform access to information in complex, dynamic networks (e.g. [8][11][15]) Information agents work on behalf of their users, extracting relevant information from heterogeneous and distributed information sources [12]. With the help of common conventions on the semantics of the information, i.e. ontologies, agents are able to fuse pieces of information in different syntaxes or semantics into a coherent view [9]. Combining ontology-based knowledge representation with the proactive interaction mechanisms of agents makes it possible for the system to adapt to changes in the information, the environment, or the preferences of the user.

Most of the efforts for utilizing MAS technology in process automation have been aimed at flexible control capable of adapting to e.g. changes in production or fault situations [5][13][14][20][21]. Even though the information handling challenges described in the above are also relevant in process automation, information agent applications for the domain remain very few (e.g. fault tolerant monitoring and diagnostics, or reactive monitoring and alarm handling). In principle, information processing of process automation would clearly benefit from MAS technology. The agent-based approach could even enable completely new, sophisticated services, supporting operational state management or alarm handling. The challenge, how-ever, is to actually define the nature and functioning of these services. The lack of suitable ontological definitions is also a setback; little effort has been put to creating semantically rich ontologies for the domain.

III. Multi-agent system based architecture for process automation information access

The multi-agent system architecture presented here aims to provide a basis for building flexible and versatile monitoring services for process automation. These agent-based services rely on efficient access to all process-related information. Information agents provide an extension to conventional process automation systems; an intelligent, active layer assisting human users in supervisory tasks. The objective is to provide a connection between all the levels of modern process automation, from physical instrumentation to enterprise-level systems.

A. Agent-Augmented Architecture

The multi-agent system discussed in this paper is designed to be an extension of the ordinary process automation system. It is a separate layer on top of the physical control system, at the same level with present-day supervisory control software, and below plant management and business systems. Fig. 1 illustrates the relationship between the information agent platform and the levels of conventional process automation, clarifying the separation of the information services from process control. Separating agent technology from real-time process control gives us the opportunity to experiment with present-day agent implementation tools.

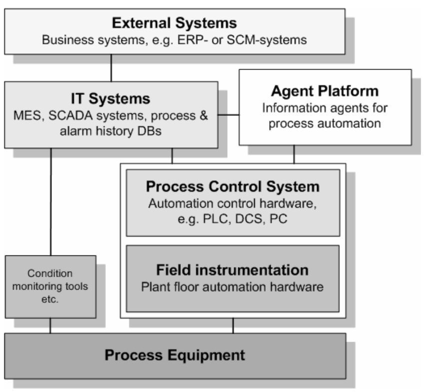

Fig. 1. The agent-augmented architecture illustrated. In this design, the information agents are located inside the Agent Platform.

The information access services can be used by both humans and computational systems. A user interface for human users can be generated by the agent system, or visualized using common process automation SCADA (Supervisory Control and Data Acquisition) software. The proposed architecture also makes it possible to format and relay information from the process control system to external computational systems.

B. Agent society and different roles of agents

The agent society consists of a set of information processing agents operating in different roles, as illustrated in Fig. 2. The setup of roles is based on a wide range of responsibilities and tasks of automation system components. The different agent roles ease application design; there is a natural place for each service or data in the agent society. Currently, there are five different roles: *Process Agent, Information Agent, Client Agent, Wrapper Agent* and *Directory Facilitator.*

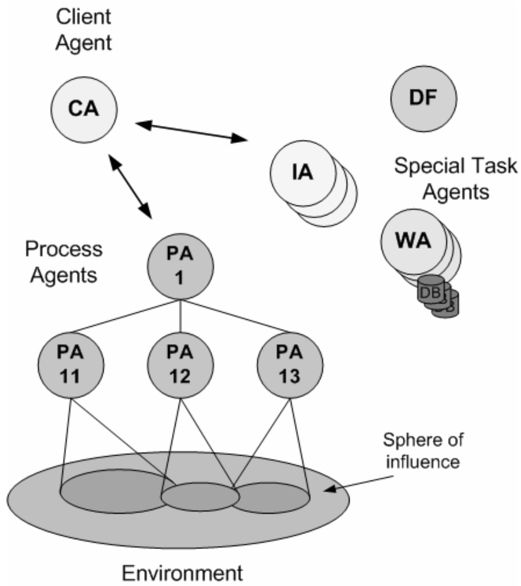

Fig. 2. A society of information agents in different roles.
CA=Client Agent, IA=Information Agent, WA=Wrapper Agent, DF=Directory Facilitator, PA=Process Agent

The key agent type is the *Process Agent*, which is responsible of a certain spatially or functionally divided process area. It actively monitors the process variables, and watches over changes in the physical system setup. The *Process Agents* form a hierarchy, handling data abstraction from vertical level to another. At the lowest process level agents handle detailed measurement information, while agents on the higher levels deal with shorter, more abstract summaries about the process conditions.

An *Information Agent* is a specialist that provides various information processing services for other agents. The services can include e.g. information searching or data formatting. An *Information Agent* has the ability to decompose information related goals to subgoals. These subgoals are then passed to other agents, which have the possibility to fulfil them. In other words, *Information Agents* have the ability to delegate information tasks to other agents [1].

A *Client Agent* provides an interface for human users, and translates human understandable queries to agent communication language. *Wrapper Agents* are used by other agents to access information stored in legacy data sources. *Wrappers* also translate data to common representation format, and may provide basic data mining services. Finally, the *Directory Facilitator* (DF) agent provides a wide variety of directory services for agent discovery (The DF is standardized by FIPA [7]).

C. Goal-oriented functionality

The operation of the agents is based on a goal-orientated principle, which has its foundation in the BDI-model (Belief-Desire-Intention [1][19]). Inside an agent, there is a manager module that controls the use of individual information processing modules. Fig. 3 illustrates the internal design of a general information processing agent. In the architecture, there is a clear separation between goal-based control over actions and the particular actions for information processing.

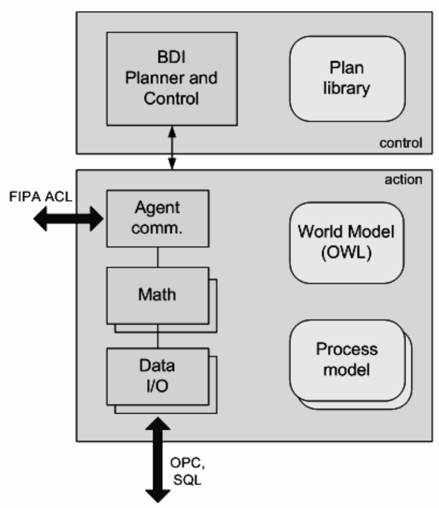

Fig. 3. The internal design of a BDI-model based information agent. (Figure is subject to change.)

One of the key motivations for adopting the goal-oriented approach is that in automation, the details of information processing depend on the context. For example, certain data algorithms or process models cannot be utilized in abnormal process states. BDI-based planning systems are argued to be capable of dynamically adapting to changing environmental situations [6]. In addition, when using goal-oriented engineering, the user may be left ignorant of the details of information processing. Furthermore, goal-oriented operation design makes it possible to use concepts that are intuitive for human users [1][19].

D. Ontologies

The use of a common ontology provides the user and the agent society with a transparent access to information. As the wrapper agents translate and publish the information in the ontology-based format, the receiver can access the information without knowing the original source, syntax, or semantics. On the other hand, a formal conceptual model of the domain assists the automatic processing (combining and reasoning) of information on different levels of abstraction.

In process automation, a common representation of knowledge is needed to bind together the mass of heterogeneous process information. There are many standards for information representation in the field, but the agents need to be able to address both low-level (e.g., numerical process measurement data) and high-level (e.g., symbolic operational state definition) information within the same context.

In our architecture, the common ontology describes (a) the concepts of the process dynamics, control, and automation to facilitate the interoperability of different process models, (b) the concepts of process information exchanged between the different IT systems to support information integration, and (c) the services provided by the agents. The ontology is written in OWL (Ontology Web Language, [17]), the current W3C standard for ontological modelling. At current, our ontology is quite limited, as it is only intended to serve our demonstration purposes.

E. Information processing and accessing modules

The actual information processing is performed by modules that are specialized in certain functions (see Fig. 3). For input and output of data, there are *data IO* modules, which communicate with legacy information systems and translate data to the common ontology. Communication between actors in the agent society is handled by an *Agent communication* module, which also is responsible for transforming data from the internal representation format to be suitable for agent negotiations. For actual information processing and reasoning purposes, an agent has a number of *Math* modules. These modules are responsible for information processing tasks.

IV. PROCESS MONITORING OPERATIONS

To enable efficient process automation monitoring tasks, information from versatile data sources must be combined in a flexible manner. Usually, a combination of different information is needed to be able to throw conclusions about the overall state of the process. Some of this information is by nature available and continuously changing (process measurement data). For other kind of information, occasionally a long time has to be waited before a triggering event occurs. Although there already are many industrial strength monitoring and diagnostics solutions available, there is no common framework that could be used for combining these partial solutions effectively. This framework should provide a starting point for monitoring applications, similarly as SQL servers do in knowledge industry.

There is a strong need for more powerful monitoring applications in the future. These applications should be easy to design, configure and maintain. One could argue that the building blocks are available, but the framework enabling the connection of these blocks is not. Using conventional software philosophies to execute the connection is laborious and may result in a solution that is hard to reuse.

We argue that the agent based architecture presented in Section III is potentially a generic and feasible solution to this problem. The architecture provides a basis for building advanced process automation monitoring applications. In this section, further details on how to build monitoring applications using this architecture are presented. The section ends with discussion about potential monitoring applications.

A. General information processing operations

Before advanced applications can be built, general basic operations for information processing have to be defined. Currently, we divide these basic operations to three types: integration, monitoring and reasoning. Although the division is not meant to be final, we have found it to be a useful starting point.

First, there is a need to integrate information generated over the physical instrumentation of process automation. Integration components are needed to provide access to different data sources that rely on physically distributed systems. These components must convert data to a shared, commonly understood format. There are many applicable technical solutions to achieve this. In our approach, we use agent-based techniques to search and retrieve data; wrapper agents that access legacy data systems, and directory services. For data formatting, we claim that ontology-based translation is the appropriate methodology [3][16].

Secondly, in automation, monitoring of various directly measured or mathematically derived values is important. Characteristic to monitoring operations is that it is not known in advance when a deviation will occur. There might be long periods of time until a triggering event takes place, and the interdependency of process variables may cause a flood of notifications in a fault situation. Previously used cyclic (or polling) operation provides no means to react in timely fashion to changes in the process and in its setups. However, agent-based methods enable us to define many active, flexible participants in process monitoring. This way, decision making and monitoring can be situated near the source of the information. Also, various interaction protocols have been standardized for agent communication, and these are argued to provide a means to manage asynchronous information flow. For the purposes of process automation monitoring, the subscribe interaction protocol is particularly interesting [18].

Thirdly, to be able to throw some conclusions about process conditions, we need to reason over available data. This reasoning produces new knowledge from given raw data. Reasoning may include the use of various mathematical methods, or the utilization of different types of process models. In the past, there have been a very large number of research activities aimed at developing mathematical methods for condition monitoring and fault diagnostics. Therefore, it can be argued that there is a lot of expertise available in this area that is ready to be adopted. Nevertheless, great effort is needed to configure and combine these methods and tools so that their full potential can be achieved.

These three basic operations cannot by nature be integrated in the same way as normal software components. Instead, we need a particular framework and design methodology to support the building of applications on top of these operations. Next, we present that goal-oriented operation principle can be used to handle this issue.

B. Building monitoring applications

It is easy to give instructions on how to perform a monitoring task to a human, but there is no ready agreement on how this could be done with computational systems. For physical control operations, there are a large number of standards for expressing control logic, for example the IEC 61311-3 programming language for PLCs. For monitoring services, such languages or methodologies have not been proposed. Based on our architecture, we propose goal-oriented solution as a starting point. Fig. 4 shows an example on how monitoring tasks could be partitioned and delivered as high-level goals between actors in a monitoring network. In general, the agents request each other to fulfil various goals that represent monitoring tasks.

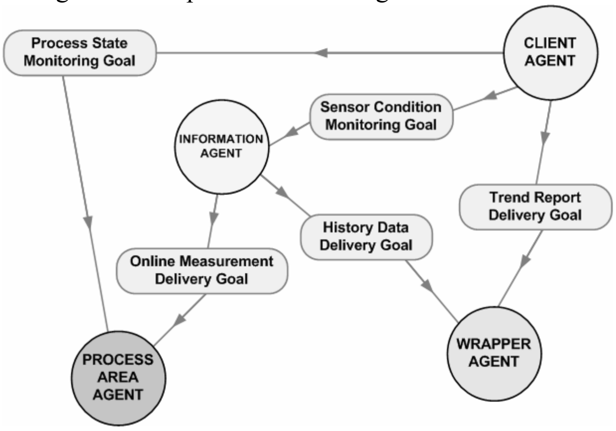

Fig. 4 An exemplar monitoring system with multiple actors requesting others to achieve different goals related to process monitoring.

Using goals as elementary components for describing the monitoring services provides functional flexibility that cannot be achieved with other operational principles. For example, when passing requests in traditional client-server solutions, the response is assumed to be provided instantaneously, which is not always possible. Furthermore, invoking functions remotely binds the service provider to use certain means to accomplish what is wanted. However, passing goals enables agents to decide locally *how* and *when* to acquire the requested information. After all, the other agents are interested in the information that is the outcome of monitoring, not in the methods used to generate the information [1][8][12].

When an agent receives a goal from its colleague, it may not be able to fulfil it alone. If the goal concerns things the agent knows itself, the agent answers directly or starts local operations that generate the needed outcome. An example of such direct operation is *Process State Monitoring Goal* shown in Fig. 4. If the goal is about things that the agent can only generate by processing information acquired from elsewhere, it transmits partial information goals to other participants. For example, in Fig. 4, information agent partitions the *Sensor Condition Monitoring Goal* and passes two separate subgoals (*Online Measurement Delivery Goal* and *History Data Delivery Goal*) to its peers.

In the design and implementation phase the concepts used to specify and describe the goals are selected from a monitoring ontology. In the future, ontologies that explicitly define monitoring tasks and related process concepts should be publicly available. Currently there are no standardized monitoring ontologies, and the examples introduced here are only demonstrative. An explicitly defined terminology is crucial for agent-based cooperation, and it also provides a foundation for error-free communication between agents and humans.

Theoretically, agent-based and goal-driven monitoring seems to be an applicable and attractive solution. However, a systematic methodology is needed before any larger scale experiments can be executed. For example, Tropos [2] is a proposed design framework for agents with goal-oriented operation principles. Although it provides a solid basis for goal-oriented design, and has been used in this paper to present our designs, it is not intended to cover issues related to condition monitoring.

Naturally, there are a lot of issues in process automation that are not covered by common agent design tools available now or in the foreseeable future. Therefore, if agent-based solutions are to be applied, a variant of a common agent design methodology has to be developed. This methodology should be able to describe various monitoring tasks and handle diverse data formats. It should be able to incorporate information from automation engineering documents, and cover numerous mathematical procedures.

C. Vision of Potential Monitoring Services

The motivation to build the architecture described in this paper is to enable new kind of monitoring services for process automation. We argue that the architecture makes it possible to link and integrate existing partial monitoring solutions in a new way. The approach could produce totally new kinds of integrated services to human users as well as external computational systems. For example, the agent-based system could answer to complex condition monitoring questions like "*is there any PT100 type of temperature measurement oscillating during process start*", or "*notify if some online measurement differs from the corresponding laboratory measurement*".

All the information needed to answer these kinds of questions is already available, but with existing automation systems, the necessary reasoning requires human deduction. Process operators use different user interfaces and views to search and gather information from multiple databases. This supervising is the kind of work that process operators and service personnel do every day. However, as the monitoring tasks are repeated from day to day, human users easily lose interest and attention. The basic idea is that the performance and reliability of monitoring would considerably be increased if some of the repetitive work could be automated using modern ICT tools. In addition, an increasing number of targets could be traced with automatic supervision.

We believe that in the future, advanced monitoring services rely heavily on combining various pieces of process-related information from different sources. For example, fusing device diagnostics with process state analysis and process models would make it possible to eliminate some of the false alarms that currently confuse and annoy supervisory staff. In our vision, the potential monitoring services are multifunctional and reliable; when they identify a problem in the process, there really is one.

V. DEMONSTRATIONS

In order to test the feasibility of the presented architecture, we have used demonstration settings motivated by real world challenges. The demonstration scenario is an actual mechanical pulp bleaching process from paper making industry. Typically this application has high accuracy requirements against important process measurements, and the instrumentation used is situated in a hostile environment [10]. Inaccuracy and malfunctions, originating measurements that are drifting or false, should be avoided in every case.

Process industry uses a lot of online sensors that typically derive a measurement on the basis of some known chemical or physical phenomenon. The sensors are usually complex by structure, require careful calibration, and therefore are subject to faults. Incorrect measurements can cause severe problems in process control. Against this setting, we have carried out a test scenario that focuses on the validation of an online measurement. In this test, an uncertain online measurement is compared to an exact manual laboratory measurement.

A goal-oriented design of this test is illustrated in Fig. 5. The user is interested in the condition of a sensor, and uses the *Client Agent* to submit a monitoring task to an *Information Agent.* As the Information Agent aims to compare online and laboratory measurements, the first task is to search for agents that can provide this information (*Information Source Search Goal*). After the relevant sources have been found, the agent passes the appropriate requests (*Lab. Measurement Delivery Goal* and *Online Measurement Delivery Goal*).

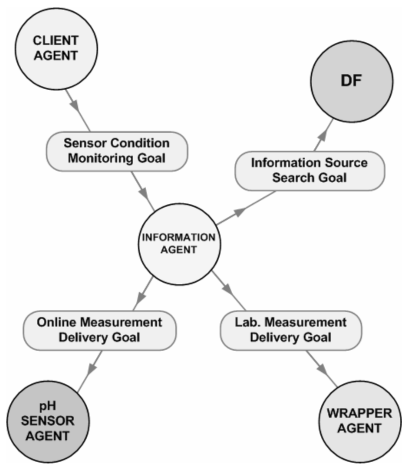

Fig. 5. A demonstrative example: a number of cooperative information agents monitor physical operation of pH measurement sensor.

The detailed description on how the *Information Agent* operates upon receiving the *Sensor Condition Monitoring Goal* is not presented in this paper. However, in general, all monitoring tasks are represented by plans, and items in plans refer to concepts in the ontology-based process model. In this test setup, the process model indicates that a laboratory database system provides the exact values of a particular process variable at certain time intervals. As the same process variable is measured by the sensor under consideration, these two values may be compared to conclude the condition of the sensor device.

The demonstration was carried out by building an agent system conforming to the architecture discussed in Section III, using principles handled in Section IV. The scenario is actually a partial solution to the second advanced monitoring question presented in Section IV C. Therefore, we argue that the presented guidelines may be used to build next generation process automation monitoring services. However, there are more open questions than there are answers provided here about the detailed operation. In the future, more complex test setups are needed to verify the applicability of the proposed techniques and procedures.

VI. CONCLUSIONS

In this paper, we presented an agent-based, goal-oriented approach for advanced process automation condition monitoring. The motivation for this research is the possible match between the features of goal-oriented agents, and the needs of condition monitoring. Our research shows that goal-oriented design and engineering methods can be used to construct condition monitoring services. However, a large number of technical details are still unresolved. Nevertheless, a demonstration motivated by a real world challenge, the validation of the correct operation of an online measurement device using a manual laboratory measurement, has been successfully implemented according to our approach.

In the future, the presented approach should be tested with more complex condition monitoring tasks. Future research focuses on the detailed design of the condition monitoring services, and further test scenarios motivated by real world challenges. Furthermore, the development of formal engineering methods to support the design of agent-based condition monitoring services is under consideration.

VII. REFERENCES

[1] L. Braubach, A. Pokahr, D. Moldt, W. Lamersdorf, "Goal Representation for BDI Agent Systems", in *The Second International Workshop on Programming Multiagent Systems* (PROMAS-2004)

[2] J. Castro, M. Kolp, J. Mylopoulos, "Towards requirements-driven information systems engineering: The Tropos project", Université catholique de Louvain, Institut d'administration et de gestion, Working paper 31/02

[3] M. Ciocoiu, D. Nau., "Ontology-Based Semantics", in *the Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning*, Breckenbridge, Colorado, April 12-17, 2000

[4] D. Cockburn and N.R. Jennings, "ARCHON: A distributed artificial intelligence system for industrial applications". In *Foundations of Distributed Artificial Intelligence*, G.M.P. O'Hare, N.R. Jennings NR, ed.: Wiley & Sons, 1995.

[5] S. M. Deen (ed.), *Agent Based Manufacturing: Advances in the Holonic Approach,* Springer-Verlag.

[6] M.E. desJardins, H.D. Durfee, C.L. Ortiz Jr., M.J. Wolverton, "A Survey of Research in Distributed, Continual Planning", *AI Magazine*, 4, 13-- 22, 1999.

[7] FIPA. The Foundation for Intelligent Physical Agents, http://www.fipa.org, 2005.

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

[9] M. Klusch, "Information Agent Technology for the Internet: A survey". *Data & Knowledge Engineering*, Elsevier, Vol. 36, 61-74, 2001.

[10] K. Leiviskä. "Process Control", *Vol 14 of the Papermaking Science and Technology Series*, TAPPI, ISBN: 952-5216-14-4, 1999.

[11] M. Luck, *Agent Technology Roadmap: Overview and Consultation Report, Agent based computing*, Agentlink III, December 2004, downloaded 28.2.2005, http://www.agentlink.org/roadmap/

[12] P. Maes, "Agents that reduce work and information overload", *Communications of the ACM archive*, Volume 37, Issue 7, 1994, Pages 30 - 40.

[13] V. Marík, M. Fletcher, M. Pechoucek. "Holons & Agents: Recent Developments and Mutual Impacts", in *Multi-Agent Systems and Applications II*, V. Marik, O. Stepankova, H. Krautwurmova, M. Luck, ed. Springer-Verlag, Germany, pp. 323-335, 2002.

[14] V. Marík, D. McFarlane, "Industrial Adoption of Agent-Based Technologies", *Intelligent Systems,* Volume: 20, Issue: 1, Year: Jan.-Feb. 2005, pages 27- 35

[15] M. Nodine, A.H.H. Ngu, A. Cassandra, W.G. Bohrer, "Scalable Semantic Brokering over Dynamic Heterogeneous Data Sources in InfoSleuth™", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 5, 2003.

[16] M. Obitko, V. Marik, "Adding OWL Semantics to Ontologies Used In Multi-Agent Systems for Manufacturing", in *Proceedings of the 1st International Conference on Applications of Holonic and Multi-Agent Systems* (HoloMAS 2003), 2003.

[17] OWL Web Ontology Language, http://www.w3.org/TR/owl-ref/, 2004.

[18] T. Pirttioja, I. Seilonen, P. Appelqvist, A. Halme and K. Koskinen, " Agent-based architecture for information handling in automation systems", in *6th IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services* (BASYS'04), 2004.

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

[20] I. Seilonen, P. Appelqvist, M. Vainio, A. Halme, and K. Koskinen, "A concept of an agent-augmented process automation system", in *17th IEEE International Symposium on Intelligent Control* (ISIC'02), Vancouver, Canada, October 27-30, 2002.

[21] I. Seilonen, T. Pirttioja, P. Appelqvist, A. Halme and K. Koskinen, "Cooperating subprocess agents in process automation," in *Proceedings of the 1st International Conference on Applications of Holonic and Multi-Agent Systems* (HoloMAS 2003), 2003.

[22] T. Tommila, O. Ventä, K. Koskinen, "Next generation industrial automation – needs and opportunities", *Automation Technology Review 2001*, VTT Automation, 34-41, 2001.

[23] O. Ventä "Research View of Finnish Automation Industry", *Industrial Systems Review 2005*, VTT Industrial Systems, 48-55, 2005.