Indirect Process Monitoring with Constraint Handling Agents

Ilkka Seilonen, Teppo Pirttioja, Aarne Halme, Kari Koskinen
Helsinki University of Technology
P.O.Box 5500
02015 TKK
FINLAND
ilkka.seilonen@tkk.fi

Antti Pakonen
VTT Technical Research Centre of Finland
P.O.Box 1000
02044 VTT
FINLAND
antti.pakonen@tkk.fi

***Abstract* – An approach to indirect process monitoring based on a society of constraint handling agents is presented in this paper. According to this approach an operator of a process automation system can define monitoring tasks which a group of agents perform proactively. The monitoring tasks are assumed to be composites and refer to several process measurements. The purpose of the monitoring agents is to enhance the work of the operator by letting him to supervise the indirect monitoring tasks instead of following a large amount of measurement data. The monitoring agents operate as a multi-agent system consisting of agents with constraint handling capabilities. The agents can setup and execute user configured monitoring tasks cooperatively. Constraints are used as one method for modeling the monitoring logic of the agents. The approach is illustrated with a test scenario using measurement data from an industrial process.**

I. Introduction

This study is motivated by changes in the work of the operators of process automation systems and possible new approaches to react and take advantage of these changes. The amount of measurement data available to operators has strongly increased during the last decade. At the same time there is an increasing need to coordinate control operations in a larger scope. The work of the operators can be argued to have become more intensive and broader. There is also a need for the operators to be able to configure the user interface that they use when needed.

Proactive computing [26] is a developing paradigm of human-computer interaction which might be applied to monitoring in process automation in the form of indirect management [8]. Proactive behavior of the monitoring system could help the operators to cope with the increasing demands of their work. One possible method to implement process monitoring in the form of indirect management is multi-agent systems (MAS) [6][7][30]. However, if MAS is accepted as a development method for indirect process monitoring the question how to design the agents still remains.

The purpose of this paper is to present an approach for implementing indirect process monitoring functions with constraint handling agents. A MAS with goal-oriented behavior is proposed as a suitable implementation method for indirect process monitoring functions. This choice is justified by the match between indirect management and MAS. Constraints are proposed as one possible method for modeling a part of the application logic of process monitoring functions. Constraints are expected to be useful for modeling user configurable monitoring tasks which refer to relationships between several measurements or other data.

This paper is outlined as follows. Chapter II will discuss proactive computing as a model and information agents and constraint satisfaction as methods for implementing indirect process monitoring functions. A model of the monitoring MAS is described in Chapter III and the usage of constraints in Chapter IV. An illustrating test scenario of the approach is presented in Chapter V followed by conclusions in Chapter VI.

II. ENABLING TECHNOLOGIES FOR INDIRECT PROCESS MONITORING

A. Proactive Computing and Indirect Management

Proactive computing is a novel paradigm for human-computer interaction being developed within the research of user interfaces [26]. A motivation for proactive computing is the difficulties which users have when trying to react to the increasing amount of data and events in information systems. The central concept in proactive computing is to change the way users interact with computers through giving the computer a more active role. A proactive information system is expected to facilitate the work of the users by letting them to supervise the tasks of the information systems instead of actively manipulating each of them.

The principles of proactive computing have been applied to some industrial applications in recent research. In a study concerning the user interfaces in process control systems a framework for indirect management in process control was presented [8]. According to this framework, indirect management can be applied to many supervisory functions of process control e.g. monitoring, disturbance control, information exchange and knowledge management. In another research the possibility of applying proactive computing to information systems of maintenance work was studied [22]. These studies seem to suggest the applicability of proactive computing and indirect management functions in industrial applications.

B. Information Agents

Information agents are one application area of MAS in which the task of the agents is to help human users in accessing data that they need in their activities [9][11][25]. The functions of information agents have been proposed to include e.g. planning of information access operations, locating data from various sources, translating different data representations and interpreting intermediate results [10][15][16][24]. Information agents are expected to be able to perform their tasks in a semi-autonomous way in cooperation with each other. Due to this, agents are also expected to change the way how their users interact with computers. As a conclusion, information agents can be regarded as an implementation technique for proactive computing and indirect management.

Monitoring of industrial processes is one possible application domain for information agents. Some research results have already been reported in this area. Information agents have been applied to notification handling in order to improve the awareness of the operators about process events [2][3]. In another study agents have been used for combining data from several web services for a rocket launch monitoring application [12][13]. In this study the web services provided different monitoring functions. The combination of these services was enabled with semantic models of the services and agent planning capabilities. A similar agent model has also been proposed as a general framework for applications that combine data from several sources in process performance and condition monitoring tasks [19].

C. Constraint Satisfaction

Constraint satisfaction problems (CSP) are a well known type of problems studied in computer science [23][27]. A CSP consists of a set of variables with value domains and a set of constraints which the values of the variables must satisfy. The problem in a CSP is to specify the values that satisfy all the constraints. The solution procedures of CSP combine search procedures with constraint propagation, i.e. pruning of values which can not be part of the solution. Consistency checks of the constraints are a part of the solution procedures.

The methods of constraint satisfaction problems have to some extent been applied to the tasks of process control in earlier research. In one study fuzzy constraints were used for modeling several aspects of a hydraulic system, e.g. physical structure and process dynamics [28]. Using this model, control actions can be evaluated with the techniques of constraint satisfaction and fuzzy reasoning. In another work, constraints were used for implementing qualitative modeling of a continuous process [29]. The model was then applied for verification and validation of discrete control operations.

III. MONITORING AGENTS

A. Multi-Agent System for Monitoring

MAS are proposed in here as a general implementation method for indirect process monitoring functions. The monitoring agents presented in this paper are expected to operate in conjunction with other information and automation systems used in process automation (see Fig 1). The monitoring agents are an additional part of the automation system providing services mainly for the operator user interface. The purpose of the monitoring agents is to help the operator when he needs to access data about the production process from various different sources e.g. the process control system, manufacturing execution systems (MES), computerized maintenance management systems (CMMS) and laboratory information systems (LIMS). What data is actually accessed depends on the applications designed for the agents and the availability of different information sources at a particular production site.

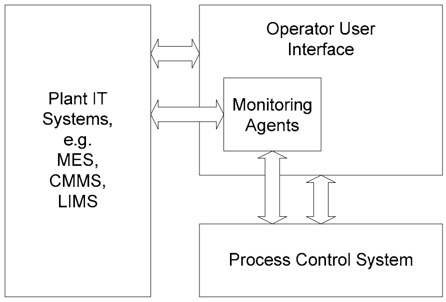

Fig. 1. Monitoring agents and their relations to other systems of process automation.

The organization of the monitoring agents consists of a few agent types with separate roles as illustrated in Fig 2. The Client Agent is a user interface agent which provides the user with functions for configuring composite indirect monitoring tasks and following their execution. Process Agents are basic monitoring agents which perform local monitoring tasks within their process areas. Information Agents are intermediate monitoring agents which operate between Client Agents and Process Agents. They decompose composite monitoring tasks to Process Agents and coordinate their execution. In addition to the mentioned agent types, also a Directory Facilitator and Wrapper Agents are needed. The Directory Facilitator is used for registering services of other agents. Wrapper Agents are used as interfaces to external information systems.

The operation of the monitoring MAS consists of setup and execution of indirect process monitoring tasks. The setup of a monitoring task is initiated by a Client Agent with which the user defines a composite monitoring task. The definition of the composite task is then passed to a suitable Information Agent which decomposes it to local tasks of Process Agents and Wrapper Agents. These agents perform their tasks concurrently and inform the Information Agent about those situations which might require coordination. The Information Agent then decides if the user needs to be informed or if the monitoring task just needs to be updated and continued. The composite monitoring tasks can be terminated according to user defined events.

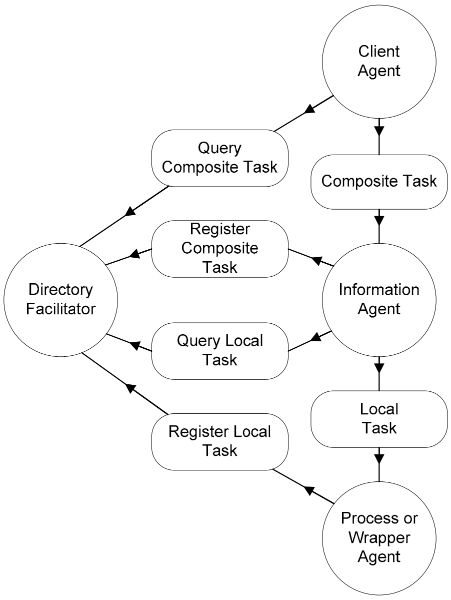

Fig. 2. Agent types in the organization of the monitoring agent society and goal exchange between them.

B. Information Agent

The most important agent type in the monitoring MAS is the Information Agent. The architecture of this agent type is illustrated in Fig. 3. This architecture combines the FIPA standard [4] as a model of agent cooperation and the BDI-architecture [1][21] as a model of a single agent. The agents communicate with each other with messages conforming to the interaction protocols defined in the FIPA standard. According to the BDI-model the Information Agent has an interpreter, a goal stack, a belief base and plans. In addition to these, the agent also needs a set of data input and processing modules. An implementation of the Information Agent was done with the Jadex open source agent development tool [20].

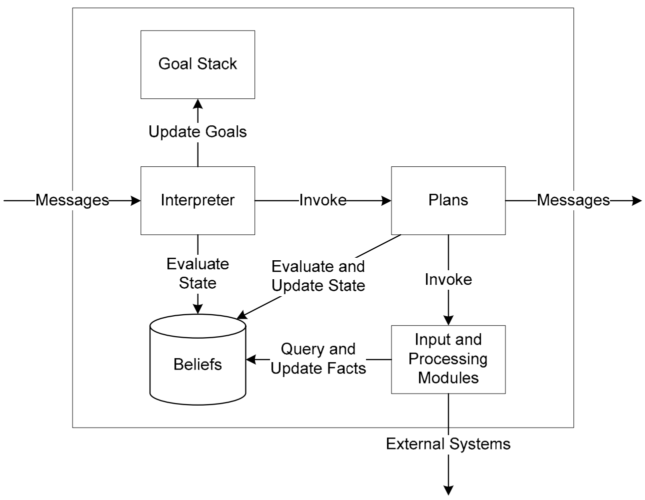

Fig. 3. Architecture of an Information Agent (modified from [20]).

The internal operation of an Information Agent is orchestrated by the interpreter which activates plans according to the current status of goals and beliefs. For each monitoring task of the agent one or more plans are defined. They specify how the monitoring task can be performed as a combination of data input, processing and communication actions. Particularly, the plans specify how the decomposition and update of composite tasks are performed during the monitoring setup and execution phases and when the user should be informed.

The architecture of an Information Agent provides a new model for developing monitoring applications in automation. The applications are defined as plans which can utilize the capabilities of the available data input and processing modules and the data in the belief base. It is envisioned that the data model of the belief base and the interfaces of the data input and processing modules can be designed fairly independently from any particular plans. Utilization of so-called ontologies might be useful for this purpose [17]. The presented architecture is expected to enhance efficient development of indirect monitoring applications in automation.

IV. CONSTRAINT HANDLING

A. Modeling Monitoring Tasks

Constraints are proposed in here as one possible method for modeling a part of the monitoring functions. Constraints are expected to be useful for modeling monitoring tasks which refer to relationships between a group of measurements or other data. The relationships are aimed for modeling operator’s knowledge of reasonable behavior of the target process. However, in the actual monitoring work this modeling scheme needs to be combined with other methods of process monitoring.

The method for modeling constraint-based monitoring tasks consists of unary and binary constraints that refer to process measurements and combinations of these with logical and-operator. The operands of the constraints may be continuous or discrete numerical variables or Boolean variables. The operators of constraints can be arithmetic or Boolean operators respectively. The comparison operators with numerical operands can express equality or inequality.

The presented form of the constraint-based modeling method has some obvious limitations. The number of variables in constraints is limited to two and only logical and-operator is allowed to combine the constraints. Overcoming these limitations with properly designed extensions is a possible part of future developments.

B. Distributed Management of Constraints

The monitoring agents manage the monitoring tasks expressed as constraints at two separate levels. The Information Agents handle composite monitoring tasks which they receive from the users. In these tasks the constraints can refer to any measurements available for monitoring. The Process Agents instead handle local monitoring tasks which are evaluated with actual measurements each time when new data is available. In these tasks the constraints can refer only to those measurements which are supervised by the particular Process Agent. The local monitoring tasks are derived from the composite tasks by the Information Agents in such a way that they can be performed concurrently by different Process Agents.

During the monitoring setup phase the Information Agent decomposes each composite monitoring task into a set of local tasks. The decomposition is based on the information about which Process Agent monitors each of the measurements in the constraints. Binary constraints referring to measurements of different Process Agents have to be decomposed into two derived unary constraints. How this decomposition is done depends on the types of the operands and operators of the constraints in the following way.

- *Constraints with numerical operands and greater or less than comparison operators.* The evaluation of these constraints can be partially decomposed. The Information Agent needs to calculate two derived unary constraints which guarantee the consistency of the original binary constraint. A basic way to calculate the boundary values for the derived unary constraints is to find a point in the measurement pair space which fulfills the equality constraint and minimizes the distance to the current values of the measurements. For simple arithmetic operators ('+', '-', '*' and '/') and second order polynomials this is a quite straightforward task.
- *Constraints with Boolean operands.* The evaluation of these constraints can be partially decomposed. Depending on the type of the Boolean operator ('and' or 'or') the Process Agent monitoring the derived unary constraint has either to inform the user directly or ask the Information agent to evaluate the original constraint.
- *Constraints with discrete numerical operands and equality comparison operators.* The evaluation of these constraints can not be decomposed. The Process Agents need to inform the Information Agent each time the values of the measurements have changed.

During the monitoring execution phase the Process Agents look for possible constraint violations. If the violated constraint belongs to the original composite monitoring task the user is informed. If the violated constraint was created during the task decomposition it is not necessarily known if the original constraint is really violated. The Process Agent has to inform the Information Agent which then evaluates the original constraint. If the original constraint is violated the user has to be informed. In the opposite case the Information Agent needs to calculate new limit values for the derived constraints and send them to the respective Process Agents.

V. ILLUSTRATING SCENARIO

The presented approach for indirect process monitoring has been demonstrated with test scenarios taken from the monitoring activities in bleaching of mechanical pulp in a paper mill. The scenarios concern about the pH control in the bleaching process, which is an important control task affecting the quality of the pulp. Malfunctions of pH sensors can cause problems which remain unobserved due to compensating control. However, this kind of a problem can be noticed by measuring the flows of sodium hydroxide before, and sulphur dioxide after the bleaching process.

The illustrating scenario presented in here contains modeling of the monitoring logic of an operator as a constraint and its execution as an indirect monitoring task with monitoring agents. A rule of thumb about the acceptable flows of sodium hydroxide and sulphur dioxide can be modeled as a binary constraint according to Eq. 1.

$$C1: \quad v_1 - v_2 > 0 \qquad (1)$$

v_1: flow of sodium hydroxide, l/s
v_2: flow of sulphur dioxide, l/s

The monitoring setup phase of the test scenario is illustrated in Fig. 4. The operator defines the monitoring constraints with the help of the Client Agent which then passes the constraint to an Information Agent identified with the help of the Directory Facilitator. Similarly, the Information Agent identifies the Process Agents supervising the measurements in the constraint with the help of the Directory Facilitator. In the end, the Information Agent queries the current values of the measurements, checks the consistency of the constraint and informs the Client Agent about the result.

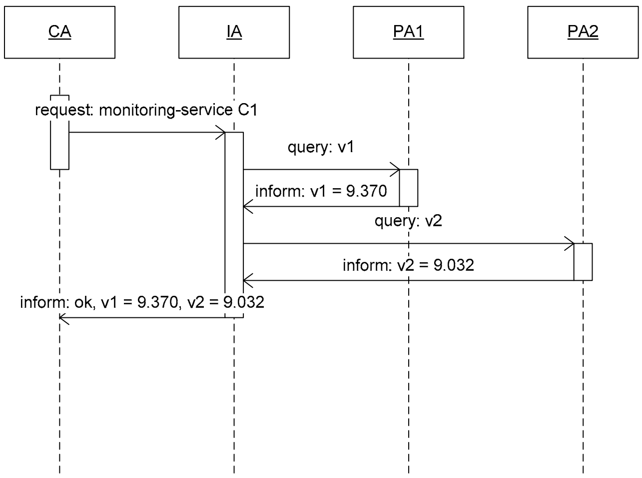

Fig. 4. Example conversation among the agents when setting up a monitoring task. The Directory Facilitator is excluded due to clarity.

The constraint decomposition phase of the test scenario is illustrated in Fig. 5. This phase is started after a confirmation from the user. The Information Agent has to initiate the monitoring tasks of the Process Agents. Because the measurements in the constraint are handled by two different Process Agents the constraint needs to be decomposed. The Information Agent calculates the following two derived unary constraints (Eqs. 2 and 3) and passes them to the Process Agents. In this case the limit value is the average of the measurement values (sodium hydroxide v_1 = 9.370 l/s and sulphur dioxide v_2 = 9.032 l/s).

C1.1a: $v_1 > 9.201$ l/s (2)
C1.2a: $v_2 < 9.201$ l/s (3)

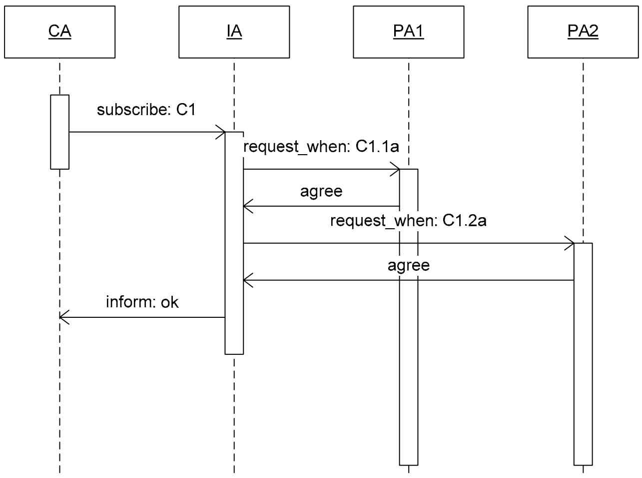

Fig. 5. Conversation among the agents when decomposing the constraints of a monitoring task.

The monitoring execution phase of the test scenario is illustrated in Fig. 6. In the presented situation the constraint of the Process Agent no. 2 is violated. The Process Agent informs the Information Agent about the situation including the current value of the measurement. The Information Agent checks the original composite constraint and concludes that it is not violated. It then calculates new limit values for the derived constraints (Eqs. 4 and 5) from the current measurement values (sodium hydroxide v_1 = 9.363 and sulphur dioxide v_2 = 9.215) and passes it to the Process Agents.

C1.1b: $v_1 > 9.289$ l/s (4)
C1.2b: $v_2 < 9.289$ l/s (5)

The presented scenario with one constraint is able to illustrate the mechanisms of constraint handling of the approach. However, the approach is expected to be more useful in more complicated cases with a larger number of different kinds of constraints. The plans of the agents that implement the constraint handling capability were designed to be independent from the content and the number of the constraints. The plans need to be modified only when new types of constraints are added, e.g. constraints with more complicated arithmetic operations. This design allows definition of the constraint-based based monitoring tasks by the end user. It is expected that an operator can perform this task with an adequately designed user interface.

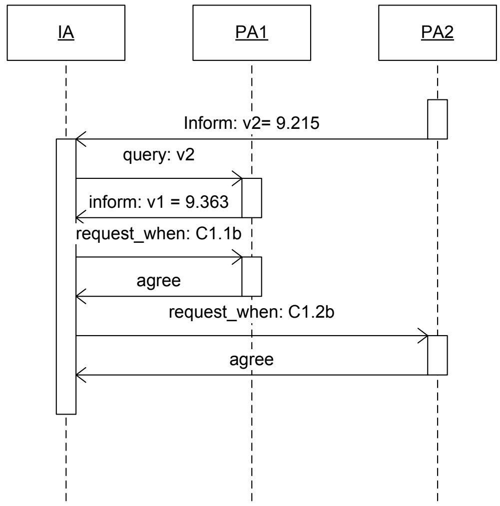

Fig. 6. Example conversation among the agents when updating the constraints of a monitoring task.

VI. CONCLUSIONS

In this paper an approach for indirect process monitoring with constraint handling agents has been presented. The approach proposes a MAS as a general implementation model and constraints as a particular modeling technique for indirect process monitoring functions. Monitoring agents are proposed as an extension to a process automation system and as an additional monitoring functionality in the operator user interface. The responsibility of the agents is to make inferences how the monitoring task should be performed. They are also expected to interpret the results obtained during monitoring and update the monitoring tasks when needed. The approach has been illustrated with a test scenario concerning pH control in pulp bleaching.

Several possible ways to extend the presented approach in further studies can be identified. The monitored information could be extended. The monitoring tasks could be developed for monitoring also the past values of the measurements. Sequences of process events could possibly be monitored in a similar way. More advanced features could also be added to the constraint-based modeling method. Fuzzy [28] and n-ary constraints combined with logical 'or' would be needed for modeling more complex monitoring tasks. Again, the constraint-based indirect monitoring tasks should be studied in combination of other process monitoring methods. Finally, the information processing in the agents could benefit from the utilization of the techniques of ontologies [18] and semantic web services [12].

VII. REFERENCES

[1] M. E. Bratman, D. J. Israel and M. E. Pollack, "Plans and Resource Bounded Practical Reasoning", *Computational Intelligence*, Vol. 4, 1988, pp. 349-355.

[2] 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", *ACM Symposium on Applied Computing*, 2004, pp. 94-99.

[3] L. Bunch, M. Breedy, J. M. Bradshaw, M. Carvalho, N. Suri, "KARMEN: Multi-Agent Monitoring and Notification for Complex Process", in *Proc. 2nd Int. Conf. on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS)*, Prague, Czech Republic, 2005.

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

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

[6] N. R. Jennings, "An Agent-Based Approach for Building Complex Software Systems", *Communications of the ACM*, Vol. 44, No. 4, 2001, pp. 35-41.

[7] N. R. Jennings and S. Bussmann, "Agent-Based Control Systems", *IEEE Control Systems Magazine*, 2003, pp. 61-73.

[8] T. Koskinen, M. Nieminen, H. Paunonen and J. Oksanen, "The Framework for Indirect Management Features of Process Control User Interfaces", in *Proc. 10th Int. Conf. on Human - Computer Interaction*, Crete, Greece, 2003.

[9] M. Klusch, "Information Agent Technology for the Internet: a Survey", *Data & Knowledge Engineering,* vol. 36, 2001, 2001.

[10] C. A. Knoblock, Y. Arens, C.-N. Hsu, "Cooperating Agents for Information Retrieval", in *Proc. Of the 2nd Int. Conf. on Cooperative Information Systems*, Toronto, Ontario, Canada, 1994.

[11] P. Maes, "Agents that Reduce Work and Information Overload", *Communications of the ACM*, vol. 37, no 7, 1994, pp. 30-40.

[12] S. Koide, S. Misono, N. Shimada, M. Kawamura and H. Nishio, "Decision Support System for Rocket Launch Using Semantic Web Services", Available: http://www.net.intap.or.jp/INTAP/s-web/data/www2005/11_Koide2.pdf

[13] S. Misono S. Koide N. Shimada, M. Kawamura and S. Nagano, "Distributed Collaboraitve Decision Support System for Rocket Launch Operation", in *Proc. Of the IEEE/ASME Int. Conf. on Advanced Mechatronics*, Monterey, California, USA, 2005.

[14] R. Murray, K. Åström, S. Boyd, R. Brockett and G. Stein, "Future Directions in Control in an Information-Rich World", *IEEE Control Systems Magazine*, vol. 23, no. 3, 2003, pp. 20-33.

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

[16] T. Oates, M.V. Nagendra Prasad and V.R. Lesser, "Cooperative Information-Gathering: A Distributed Problem-Solving Approach," *IEEE Proceedings on Software Engineering, Special Issue on Agent-based Systems*, vol. 144, pp. 72 - 88, 1997.

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

[18] M. Obitko, V. Marik, "Adding OWL Semantics to Ontologies Used in Multi-Agent Systems for Manufacturing", in *Proc. 1st Int. Conf. on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2003*, Prague, Czech Republic, 2003.

[19] T. Pirttioja, A. Pakonen, I. Seilonen, A. Halme and K. Koskinen, "Multi-Agent Based Information Access Services for Condition Monitoring in Process Automation", in Proc. 3rd IEEE Int. Conf. on Industrial Informatics (INDIN), Perth, Australia, 2005.

[20] Pokahr, L. Braubach, W. Lamersdorf, "Jadex: Implementing a BDI-Infrastructure for JADE Agents", in *EXP - In Search of Innovation (Special Issue on JADE)*, vol. 3, no. 3, Telecom Italia Lab, Turin, Italy, 2003, pp. 76-85.

[21] A.O. Rao and M.P. Georgeff, *BDI Agents: From Theory to Practice*, Technical Note 56, Australian Artificial Intelligence Institute, Melbourne, Australia, 1995.

[22] S. Riihiaho (ed.), *The Effects of Proactive Information Technology to Maintenance Work*, Helsinki University of Technology, Software Business and Engineering Institute, Research Reports 1, 2003.

[23] S. Russell and P. Norvig, *Artificial Intelligence, A Modern Approach*, Prentice Hall, 1995.

[24] N. Shaw, A. Mian, S. Yadav, "A Comprehensive Agent-Based Architecture for Intelligent Information Retrieval in a Distributed Heterogeneous Environment", *Decision Support Systems*, vol. 32, 2002, pp. 401-415.

[25] K. Sycara and D. Zeng, "Multi-Agent Integration of Information Gathering and Decision Support", *12th European Conference on Artificial Intelligence*, 1996, pp. 549-556.

[26] D. Tennenhouse, "Proactive Computing", *Communications of the ACM*, vol. 43, no. 5, 2000, pp. 43-50.

[27] E. P. K. Tsang, *Foundations of Constraint Satisfaction*, Academic Press, 1993.

[28] C. Y. Tyan, P. P. Wang, D. R. Bahler, D. R. "The Design of an Adaptive Multiple Agent Constraint-Based Controller for a Complex Hydraulic System", in *Proc. of the Int. Joint Conference of CFSA/IFIS/SOFT on Fuzzy Theory and Applications*, Taipei, Taiwan, 1995.

[29] H. Välisuo, *Model-Based Reasoning and Control of Process Plants*, Technical Research Centre of Finland, Publication 178, 1994.

[30] G. Weiss (ed.) *Multiagent Systems*, MIT Press, 1999.