OWL Based Information Agent Services for Process Monitoring

Antti Pakonen, Teemu Tommila
VTT Technical Research Centre of Finland
P.O. Box 1000, 02044 VTT
Espoo, Finland
{antti.pakonen, teemu.tommila}@vtt.fi

Teppo Pirttioja, Ilkka Seilonen
Helsinki University of Technology
P.O. Box 5500, 02015 TKK
Espoo, Finland
{teppo.pirttioja, ilkka.seilonen}@tkk.fi

Abstract

To determine the operational situation of a monitored industrial process, an operator needs efficient access to a wide range of information. Measurement data alone does not encapsulate the overall situation, but pieces of information have to be searched from different plant IT systems that unfortunately often have varying interfaces and data formats. Information agent and Semantic Web techniques address similar challenges in the context of the Internet by annotating heterogeneous data with formal semantics provided by ontology languages like OWL, and by providing human users with autonomous assistants for information retrieval. This paper presents an agent based concept for process automation that provides operators with easily configured information retrieval and monitoring services, releasing them from tedious data harvesting tasks.

1. Introduction

A process plant is a very information rich domain. Tens of thousands of measurements track the operation and performance of the process equipment. Useful information is also stored in various IT systems such as electronic diaries, maintenance databases etc. However, because of the variety of data formats and system interfaces used, looking through all the relevant information, especially in a busy situation, is simply not possible. And the fact that much of the information is free-form textual descriptions makes it even harder to search for relevant information. As a result, process operators are left with an insufficient understanding of the overall situation, sometimes leading to confusion or even misjudged actions.

Similar challenges also constrain the efficient use of the massive information content of the Internet. To deal with the plethora of data representations and the ambiguity of informal natural language, Semantic Web technologies are been developed to assign the information with machine-processable semantic annotations. Information agent technology is also envisioned to change the way information is retrieved and browsed. Instead of figuring out appropriate query strings for search engines, humans employ “intelligent” agents that can grasp the meaning of the information they have been sent out to find. Tirelessly browsing through masses of distributed information, the agents are able to figure out and assemble the best match of answers.

Because of the analogy in information processing between process supervision and the Internet in general, we have been studying the use of Semantic Web and information agent technologies in the context of industrial process monitoring. In this paper, we present an agent based architecture that augments the traditional process automation system, providing human users (e.g. process operators) with easily configurable, autonomous information retrieval and monitoring services. The services aim to increase the overall situation awareness of the users by combining information from various heterogeneous data sources using a common ontology for data representation and query.

The paper is structured as follows: Chapter 2 presents challenges in monitoring of industrial processes. Chapters 3 and 4 shortly review the basic concepts of information agents and ontologies, respectively. An agent-based architecture for process monitoring is then proposed in Chapter 5, and Chapter 6 focuses on developing a process monitoring ontology for the agent system. Section 7 presents a system demonstration based on a real-world industrial process. Finally, the conclusions are in Chapter 8.

2. Operational monitoring of process plants

Monitoring of industrial processes aims at detecting disturbances as early as possible in order to maintain efficient process functioning and minimise losses in production. When dealing with highly automated processes, monitoring is the main content of work during normal plant operation [1]. Typically the responsibility of a *process operator*, monitoring consists of selection and retrieval of relevant information, interpretation, and decision-making.

The typical system interface used by an operator is a display representing the structure of the controlled process, with signal values and trend graphs to show the key process variables. However, more than just measurement data is needed to fully understand the *operational situation* at the plant [2]. There are several IT systems that contain pieces of information about the overall situation:

- *Process diaries* are electronic notebooks for operations personnel. They can contain comments on changes in control strategy, product variation, fault situations, doubts about the equipment performance, etc.
- *Maintenance databases* contain notifications about planned and completed maintenance tasks (device replacements, calibrations etc.).
- *Measurement databases* contain the history of all measurements, and can offer different views or perform e.g. statistical analysis.
- *Laboratory Information Management Systems (LIMS)* contain manual sample values of process variables.

As many of these systems may be products of different vendors, it is often difficult to integrate them in such a fashion that the capabilities of each system can fully be exploited. In terms of monitoring, the task of information retrieval often becomes a tedious one. Particularly in busy situations (e.g. during process start up or shut down), operators simply do not have the time to browse through all the different displays and systems. Problems arise especially if the user interface of external systems or the terminology used differs to what the operators have been accustomed to.

The reason for the heterogeneity of systems is that the process industry in general is quite conservative in adopting open standards and quite careful in introducing new IT solutions. Instead of massive system overhauls, plants are updated bit by bit, resulting in a mess of system interfaces, semantics, and data formats.

3. Information agent systems

Agent technology in general aims at managing complex, distributed systems with autonomous entities called agents. Agents are capable of proactive, goal-based operation [3], and usually cooperate in dynamic agent societies (multi-agent systems, or MAS).

*Information agent*s are defined as intelligent software agents [4] that have access to *heterogeneous* and *geographically distributed* information sources, and which *proactively* acquire, meditate, and maintain relevant information *on behalf of users* or other agents [5]. They are expected to change the way humans interact with complex, information-rich systems – above all, the Internet – by providing them with intelligent assistants that can a) handle a wide range of data interfaces, representations, syntaxes and semantics, and b) search and filter masses of data to collect information relevant to their users' goals. In user interface research, the style of interaction where humans act in policy-making roles, managing and supervising intelligent computer agents that perform or even initiate tasks on their behalf has been called *indirect management* [6].

Some of the more common and illustrative information agent applications deal with travel or holiday planning - composing and scheduling e.g. a night at the town by collecting information about restaurants reservations, ongoing movies, and pubs according to the preferences of the user [7]. In the domain of process automation, information agents have been proposed for distributed monitoring, diagnostics, and alarm handling [8][9].

4. Ontologies

Agent communication takes place on the knowledge level, and thus requires an agreement on the *semantics* of the information exchanged. An *ontology* defines the explicit meanings of domain concepts and the relationships between them in a machine-processable format.

Research in the Semantic Web area has recently paved way for a widespread interest in ontologies. Semantic Web aims at developing data representation formats to support searching and interpretation of the massive amounts of heterogeneous, often informal and ill-structured information found in the Internet. Instead of current search methods based on string matching, ontology languages such as OWL [10] are being developed to annotate data with meanings "understood" by computers. Ontologies provide a formal conceptual model that supports combining and reasoning over information in different syntaxes and on different levels of abstractions.

While ontologies have been studied in agent communication before, the development of tools for ontological modelling and management in the Semantic Web area has resulted in widespread use of W3C languages like OWL, also in the domain of industrial automation [11][12].

5. Agent system architecture

We have developed a multi-agent system intended to support human users in tasks related to operational monitoring and maintenance of process plants, by providing them with autonomous, easily configurable monitoring services intended to extend their *situation awareness*. The agent system architecture has been influenced by both our earlier research of process control agents [13] as well as by general research on information agents [5][14].

In our architecture, the process automation system has been augmented with an agent platform residing on the Manufacturing Execution System (MES) layer (Figure

1). The agents extract and refine information from the underlying control hardware (e.g. PLC, DCS, PC) and from different plant IT systems. This add-on approach has enabled us to experiment with contemporary agent software without endangering real time process control.

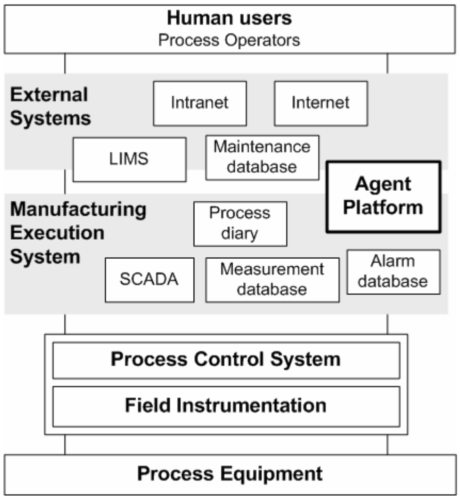

Figure 1. Process automation system augmented with an agent platform.

The agent platform is – in a number of ways – parallel to SCADA (Supervisory Control and Data Acquisition) systems. The systems exchange information, and provide the user with similar services. Other plant IT systems that the agents track are:

- Measurement and alarm history databases
- Electronic diaries
- Maintenance databases
- Laboratory Information Management System
- Corporation extranet (containing e.g. design documentation)
- External business systems (ERP, SCM)
- The Internet

All information extracted by the agents is mapped semantically to the common ontology.

5.1. Agent society roles

The agents in our multi-agent architecture act in distinct roles. The roles provide a hierarchy for agent organisation and help define the behaviour of an individual agent within the agent society. The role of an agent outlines its area of responsibility, objectives and behaviour.

Our agents have five different roles (Figure 2) that are based on the distribution of a range of responsibilities and objectives of different plant IT systems, process equipment (sensor and actuator devices), and human users. Each type of "actor" in the plant information landscape is represented by a corresponding type of agent.

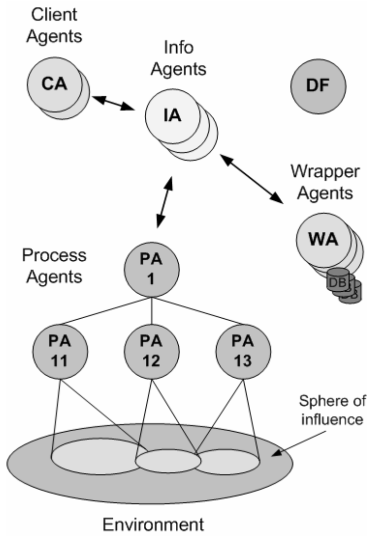

Figure 2. Agents within the agent society operate in five different roles.

Wrapper agents provide transparent access to different legacy data sources. Their tasks include 1) maintaining a data connection to the information source, 2) translating data from different formats to the ontology-based representation understood by the agent society and 3) filtering and monitoring for changes or updates in the information stored in the data source on request.

A *Process Agent* represents a section of the physical plant equipment. Process Agents are structured hierarchically, with low-level agents representing single devices such as pumps or valves. On a higher level, process agents account for plant (sub)processes, and at the top level for entire processes or plants. The agents maintain a direct connection to the hardware and employ a various set of methods in order to 1) collect and refine information about the operational state of the equipment, 2) attempt to discover deficiencies in the performance of the equipment and 3) monitor for changes in the desired information (e.g. measurements) both by default and on request.

Client Agents provide a user interface for human users in different roles (e.g. a process operator or a maintenance person). They can also be used by external systems to connect to the agent services. A Client Agent represents its user in the agent society by 1) providing the information retrieval, processing and monitoring services offered by the agent society, and 2) maintaining the conversation between the client and the agents. A separate display for agent interfacing is not required, as long as new functions can be added into existing user interfaces.

An *Info Agent* differs from the other, more or less permanent roles in that its lifespan is that of a given task. Info Agents are activated "off-the-shelf" to carry out information retrieval tasks that require the services of a number of different agents. The agents 1) decompose information retrieval and monitoring tasks to queries and subtasks for appropriate agents, 2) filter, combine and format the collected data using various methods, and 3) monitor for phenomena dependent on distributed initial data.

Finally, the *Directory Facilitator* provides a yellow pages service used by the agents to advertise their services and seek services provided by other agents. The directory service facilitates dynamic task (re)configuration and plug-and-play interaction among agents.

5.2. Agent structure

The internal structure of our agents (Figure 3) is based on the BDI (Belief-Desire-Intention) agent architecture [15]. The BDI model enables goal-based problem solving by combining the suitable courses of action (intentions) in a given situation (beliefs) to achieve adopted objectives (desires). The possible courses of action are usually depicted as *plans*.

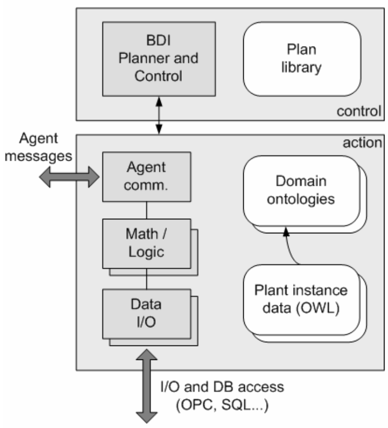

Figure 3. An agent consists of a BDI-based control unit, and several modules for information processing.

As our agents offer information retrieval services, the plans in the agent's *plan library* consist of actions necessary to extract data from different data sources, to process the data using various (symbolic or mathematic) tools and methods, and to communicate refined information to other agents. When a request is received, the *planner and controller* unit of the agent selects and performs the appropriate plan actions to achieve the requested data processing or monitoring objective [16]. If some necessary subtasks can only be achieved by other agents, the agent engages in negotiation with its peers, decomposing tasks for appropriate agents.

To facilitate transparent access to information, all data extracted from the environment is mapped to the OWL *domain ontology*, which is discussed in the next section. An agent is configured to a certain plant environment with a world model consisting of instances of the classes defined by the domain ontology.

6. Process monitoring ontology

For demonstration purposes, we developed a process monitoring domain ontology in OWL using the Protégé [17] ontology editor.

In our architecture, the domain ontology serves primarily three functions: 1) It is a formal data model needed to integrate information from heterogeneous data sources. 2) It allows us to link data (e.g. measurements) with other information that may be relevant through different contexts (time, process area, equipment type). 3) The triplet structure of OWL enables complex queries, where we can move from concept to concept quite freely (Example: "Search all calibration events x for sensors y that measure the process variable z.").

The premise for the ontology is the viewpoint of the process operator. The ontology aims at describing and structuring the domain concepts essential in monitoring activities. Pieces of information can be related to each other via the physical setup of the process equipment, or via a functional viewpoint of the activities carried out by the equipment (Figure 4).

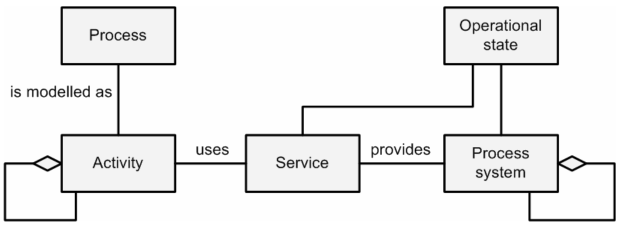

Figure 4. Concepts relevant in process monitoring: in a given operational state, physical process systems provide services needed to carry out functional activities.

The physical viewpoint embodies concepts such as devices, plants, and hierarchical process systems consisting of subsystems and devices. The functional viewpoint depicts the activities and tasks needed in transforming the raw material into end products; process phases and variables. The two viewpoints are linked by the notion of the *operational state* of equipment. In a given operational state, process systems provide services that are needed to carry out the activities necessary to manufacture the end products.

6.1. Domain ontology structure

Our ontology has been split to *subontologies* that cover different types of information sources. For example, the maintenance subontology covers the different concepts relevant in extracting information form the maintenance history database. All the subontologies contain links to the physical equipment ontology, which in a way serves as the *base ontology*. For example (Figure 5), maintenance events are related to process components such as sensor devices that are defined in the physical domain ontology. Sensors in turn provide measurements that depict process variables (such as pH) that are defined in the functional domain subontology.

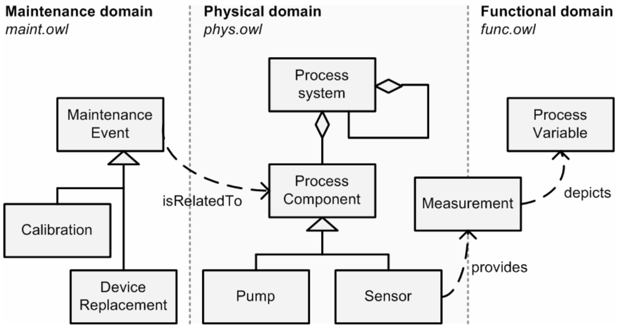

Figure 5. The concepts in different subontologies are semantically linked via the base ontology of the physical domain.

Because of this modular ontology structure, each data source Wrapper Agent can be given a "lightweight" ontological model consisting of only the subontologies needed. In another words, an agent specialised in e.g. electronic diary entries does not understand the concepts used in the maintenance domain. Using the OWL syntax, we can easily link the concepts to the base ontology, and bind all the concept definitions together.

6.2. On ontology development

In the development of the domain ontology, our intention was to capitalise on an ontology readily developed in another research project for maintenance operations of an industrial plant [12]. Although the concepts of the two ontologies were mostly the same, the different viewpoints resulted in a completely different way of structuring the relationships between the concepts. From the maintenance perspective, process devices are device individuals with a given life cycle. As all the concepts of the maintenance ontology were mapped to device *individuals*, it became almost impossible to express relationships that are of interest in the monitoring perspective. In monitoring, the interest of the operator is focused on more or less fixed process variables and measurements, while the device individuals may come and go.

As a result, the maintenance ontology was of little use in process monitoring. This serves as an example on how difficult it can be to construct universally applicable ontologies for domains where different viewpoints can be relevant for different users.

7. Demonstration

The feasibility of our agent-based architecture for process automation has been tested in demonstration scenarios [18][16] that have been motivated by an actual industrial process, a mechanical pulp bleaching plant. A real industrial case provided us with concrete, real-world monitoring challenges and demonstration scenarios, as well as actual process history data.

The objective of pulp bleaching is to increase the brightness of the end product (paper) by decolourisation of lignin, typically using an acidifying agent such as hydrogen peroxide. The bleaching process is difficult to control because of significant time delays (for the actual bleaching reaction, pulp is retained in a tower for several hours). Due to long delays, slowly accumulating faults can take hours to detect, leading to masses of spoiled product.

However, the biggest source of problems in process control is that the hostile conditions – exposure to vibrations, corrosive chemicals and wide temperature variations – lead to unreliable operation of the process equipment. Particularly vulnerable are the sensor devices that measure key process variables such as pH or brightness.

7.1. Demonstration system

For demonstration purposes, we built a tentative implementation of our agent architecture using publicly available, open-source Java tools. A BDI-based agent development tool called Jadex [19] was selected as the agent platform. Jena [20] toolkit was used for OWL processing. A browser-based user interface was created using the readily available Tomcat server interface for Jadex.

The demonstrations were conducted off-line. Database dumps from the plant measurement history, maintenance database, laboratory database and electronic diary were accessed by the agents using an ODBC bridge. An OPC connection was also constructed for possible online demonstrations in the future.

Roughly the same system implementation was used to carry out an earlier demonstration, which involved an easily configurable monitoring agent dispatched by the user to temporarily track a set of measurements [17].

7.2. Case scenario

In the demonstration, the operator is provided with an information search service, enabling him or her to look for interesting events occurred e.g. in the previous shifts. In a problem situation, the operator can also look for previous occurrences of similar problems, how they were dealt with, and what the results were. The agents then search for the events matching the user-defined criteria, and combine them with related measurements.

As sensitive sensor devices for key quality measurements such as pH or brightness can easily malfunction in the harsh environment of the plant – and the malfunctions typically manifest in slowly drifting faults that are hard to notice – operators need to constantly assess diverse information regarding the reliability of the signals. Maintenance operations often have drastic influence on measurement trends, and a recent calibration is often a sign of a reliable measurement. Manual laboratory samples usually expose drifting measurements at an early stage. The process diary might also contain vital information on the quality variables noted in previous shifts. However, because of the different system interfaces, browsing through all the aforementioned information is tiresome.

7.3. Using the Info Agent

The demonstration involves seven agents; a Client Agent, an Info Agent, a Directory Facilitator, and four Wrapper agents for the maintenance events, diary entries, laboratory measurements, and measurement trend history, respectively. An excerpt of the agent interaction involving two of the wrappers (and omitting the Directory Facilitator) is shown in figure 6. If other data sources are required in the task, similar conversation takes place with the other wrappers.

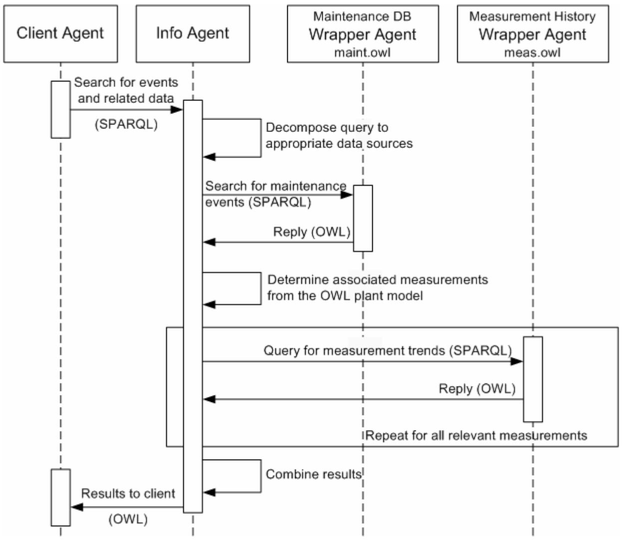

Figure 6. An example of task decomposition in the demonstration scenario.

The Client Agent provides the user with a query sheet for information search criteria. The operator can limit the search in terms of time, process area, and event type. The user-defined search criteria are passed to the Info Agent in SPARQL format [21].

The Info Agent then locates necessary data sources from the Directory Facilitator, and the query is decomposed to appropriate Wrapper Agents. First, discrete events matching the criteria are searched in the maintenance database, the electronic diary, and laboratory measurement database. If matching events are found, the Info Agent then looks for related information such as measurements from the appropriate wrapper.

Finally, the results are presented to the user in the format shown in the next chapter.

The matching of related information from different sources is based on the triplet structure of OWL and SPARQL. The SPARQL syntax allows us to encapsulate the logic behind the information composition task in declarative form. The agents supplement the original query by including predefined SPARQL fragments, directing the agents to e.g. “attach the maintenance events with measurement trends from sensors that measure the same process variable as the sensor device that was serviced” for the task in figure 6 (see also figure 5). The maintenance event is now not only supplemented with the measurement of the specific device, but also with concurrent measurements of the same variable.

7.4. Test run

In the following, we well highlight some of the more useful snapshot views to the plant information assembled by the agent society in the demonstration setting.

In Figure 7, we see a pulp pH measurement that clearly drops at about 12:50 PM. Judging from the measurement curve alone, an operator might suspect that a) the pH value of pulp is incorrect or b) the sensor device has malfunctioned. However, the measurement has been supplemented with notes of a pH quality measurement (the black dot labelled “Q”) and a maintenance database entry (the square labelled “M”).

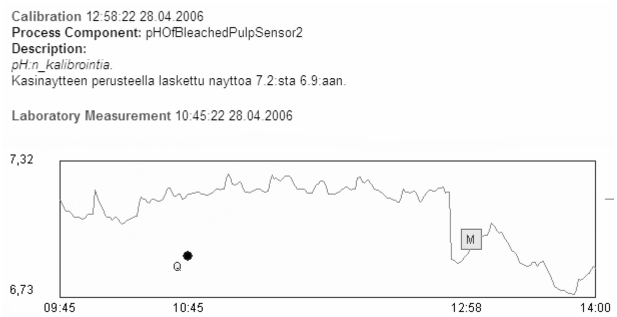

Figure 7. An online and laboratory pH measurement, and a maintenance event.

As can be seen, the quality measurement differs from the online measurement by approximately 0.3 units. This has been noticed by someone at plant, because the maintenance database entry (in Finnish) records a calibration operation done “on the basis of a manual sample”. Given all these pieces of information, the operator now interprets the situation correctly: the pH sensor was calibrated because the deviation from the value of the laboratory measurement. That is, the actual pH value did not drop at 12:50 PM.

Figure 8 shows a similar situation; a pH sensor has drifted to inaccurate values, which is noticed by comparing the measurement with a laboratory sample.

This time however, there is no notification of the calibration operation in any data source. Still, a diary entry can found that states: “online pH measurement 7.33, laboratory sample 6.89”. Judging by the dramatic shifts in the measurement curve, the pH sensor has been taken offline for the calibration, or maybe even replaced with another sensor.

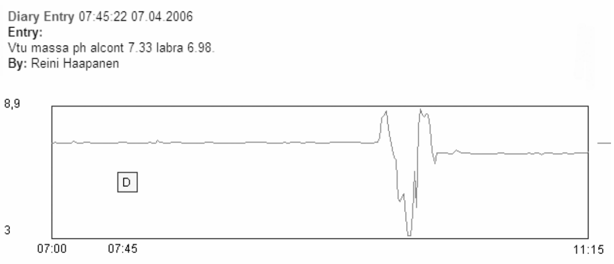

Figure 8. A pH measurement and a diary entry.

Finally in figure 9, a pulp consistency measurement exhibits a fair jump at about 2:30 PM. The shift is explained by the attached event description, which states that the measurement was calibrated to match the measurement of a concurrent pulp line. The written description also refers to a similar incident about a month ago.

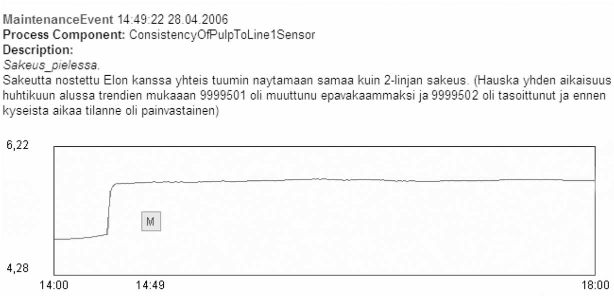

Figure 9. A consistency measurement and a maintenance event.

7.5. Further issues

The kind of markers that supplemented the measurement curves in the examples above serve as a simple verification of the measurement data. The continuous data was attached with events from various data sources, which enabled the operator to dismiss the measurements as faulty. This kind of measurement validation would also support e.g. automatic filtering of alarms, as signal shifts or peaks caused by maintenance activities could be ignored. History data could also be put into better use, if we could easily exclude unrepresentative data from data analysis. Measurement history data in itself is not very useful, unless we have stored knowledge about its context.

In order to successfully harvest information form data sources containing written information (e.g. electronic diaries) special care has to be taken in user interface development. Entries tend to be informal and ill-structured, unless the interface aims to capture explicit information by e.g. presenting a set of questions with pre-defined answer options for the users to pick. The facts that should be established unambiguously include at least time and place (that is to say which devices or process areas are concerned), but preferably also a rudimentary classification of the reported event.

8. Conclusions

In this paper, we presented an agent based system architecture for monitoring of industrial processes. The approach was to augment the traditional automation system with a layer of information agents that provide human users with information retrieval and monitoring services. These services enhance the operational situation awareness of process operators by freeing them from tedious information browsing. The agents hide the complexity resulting from different system interfaces and data formats in the various information systems at the plant.

The demonstrations presented here and earlier [18] outlined information retrieval and monitoring services that are easily configured by the user. Agent technology and OWL ontologies were used to handle the heterogeneity of various plant IT systems. The agent approach facilitates efficient process monitoring by emphasising the role of human users as decision makers by providing them with autonomous, configurable machine assistants. A major strength of our approach is that the proposed agent services can be implemented on top of existing plant IT system infrastructures.

OWL offers a versatile data format for extracting and integrating knowledge from heterogeneous data systems. The web of concepts suggested by the triplet structure of OWL enables complex, sophisticated queries to be run over distributed data. There is a lack of established query and rule languages, but this is being addressed by e.g. efforts of the W3C.

The development of generally useful ontologies for process automation is very difficult. Because of different viewpoints on how the key concepts relate to each other, ontologies made for e.g. process monitoring may be very differently structured than those made for maintenance or design engineering, even though the general domain and the key concepts are roughly the same. This example highlights the challenges in creating universally applicable process automation ontologies that could be bought and mobilised off-the-shelf.

However, as the tools available for ontology processing constantly evolve, we have found out that it is relatively easy to construct limited, “lightweight” ontologies for each application that solve only the integration challenges needs that are relevant for that specific application. Fortunately, the OWL syntax has been developed from the start to be able to address and cross-reference a set of different ontologies, instead of relying on a single monolith to solve everything.

In the future, we aim to investigate the use of ontologies and agent technology in parallel with Service Oriented Architecture (SOA) and Web Services techniques. The utilisation of the agent applications we have suggested does not require the backbone of the system to be realised as a multi-agent platform. Agents can – and we believe they will – have a specific and natural role alongside other future IT infrastructures.

Acknowledgements

Our research has been funded by Tekes (Finnish Funding Agency for Technology and Innovation), Metso Automation, and UPM. The authors would also like to thank Mr Janne Jussila for his contributions to the study.

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*, pp. 48-55, VTT Industrial Systems, 2005.

[3] N.R. Jennings, "On agent-based software engineering," *Artifical Intelligence,* Vol. 117, pp. 277-296, 2000.

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

[5] M. Klusch, "Information agent technology for the Internet: a survey," *Data & Knowledge Engineering*, vol. 36, pp. 337-372, 2001.

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

[7] G.A. Grimnes, S. Chalmers, P. Edwards, A. Preece, "GraniteNights – A Multi-agent Scheduler Utilising Semantic Web Technology", *7th International Workshop on Cooperative Information Agents*, CIA 2003, Helsinki, Finland, August 27-29, 2003.

[8] 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.

[9] 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.

[10] OWL Web Ontology Language, W3C Recommendation, http://www.w3c.org/TR/owl-features/, 2004.

[11] 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.

[12] M. Viinikkala, S. Syrjälä, S. Kuikka, "A Maintenance Demand Analyzer – a Web Service based on a Semantic Plant Model", *4th International IEEE Conference on Industrial Informatics*, INDIN 2006, Singapore, 16-18 August, 2006.

[13] I. Seilonen, *An Extended Automation System: An approach Based on a Multi-Agent System*, Dissertation, Helsinki University of Technology, Laboratory of Information and Computer Systems in Automation, 2006.

[14] 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.

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

[16] T. Pirttioja, A. Pakonen, I. Seilonen, A. Halme, 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.

[17] The Protégé Ontology Editor and Knowledge Acquisition System, http://protege.stanford.edu/

[18] A. Pakonen, T. Pirttioja, I. Seilonen, T. Tommila, "Proactive Computing in Process Monitoring: Information Agents for Operator Support", *11th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2006*, Prague, Czech Republic, September 20-22, 2006.

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

[20] Jena - A Semantic Web Framework for Java, http://jena.sourceforge.net/

[21] SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/