Semantic Business Analytics in Industrial Facilities – a Case Study

Jürgen Angele, Eddie Mönch

ontoprise GmbH
An der RaumFabrik 29
76227 Karlsruhe
angele@ontoprise.de
eddie.moench@ontoprise.de

Abstract: In the following article we point out the potentials of semantic technologies for an application in the area of business analytics. Semantic Business Analytics as understood by the operators of a power plant means to provide intelligent, experience based monitoring of the machines' sensor data, to provide interactive guidelines to advise the operator to take the right steps, and to provide a mechanism to learn from past incidents. Therefore a sensor-based ontology represents a predictive model for the behaviour of mechanical and electrical parts of a power plant under certain circumstances. Rules are crucial in this application, as they represent, as so called monitors, complex relationships between different elements (energy medium, parts of the plant, and environmental influence) and their behaviour in time. Acting as semantic agents, these monitors observe the power plant and create alerts in case of an incident. The alert is used by an ontology-based advisory system which guides the operator how to deal with and fix the problem. The knowledge engineer is able to improve the system by learning from past incidents by analyzing the stored incidents and derive new experience and monitors from. Monitoring systems may be classified as business analytics systems and thus this application shows the benefits of semantic technologies in business analytics respectively business intelligence.

1 Introduction

"Predictive analytics encompasses a variety of techniques from statistics and data mining that analyze current and historical data to make predictions about future events. Such predictions rarely take the form of absolute statements, and are more likely to be expressed as values that correspond to the odds of a particular event or behaviour taking place in the future" [WP09].

For predictive analysis semantics may play an important role. Regarding our use case, it induced several requirements to the chosen underlying model:

- The prediction models must be communicable between different domain experts and the business people.

- The terminology must be well defined and well understood. The models must be very flexible concerning modifications, thus requiring an abstract representation level.

- The models must be immediately operational without any representation breaks in between. Predictive models exploit patterns found in historical and transactional data to identify risks and opportunities. Models capture relationships, among many factors to allow assessment of risk or potential associated with a particular set of conditions, guiding decision making for candidate transactions.

We apply ontologies to define these predictive models. As the relationships are very complex, expressivity of the representation language is crucial. Declarative rules are the right means to define these complex relationships on a high level of abstraction.

Our customer is ALSTOM Power Generation AG which is under the global leaders in the world of power generation. In our use case Alstom is commissioning a hydro power plant in Malaysia. This project is called the "Bakun Hydroelectric Project". The customer of the Bakun Hydroelectric Project requested an expert system delivered with the plant, focusing on supporting the plant operators in predicting and handling issues in difficult situations.

The paper is structured as following. First we describe in more detail the requirements of the project. Then we show how we met the requirements based on semantic technologies from ontoprise. The architecture and conclusion finally round up the paper.

2 Given Requirements

During analysis the customer posed a set of requirements. Basically the system should have three functional parts. The monitoring part analyses the data and events being received from the power plant components. The guide part gives advice to deal with critical issues detected by the monitors, and finally the system should be able to learn from previous incidents. The whole expert shell system for Alstom is called CEXS (**C**omputer Guided **Ex**pert **S**hell System).

2.1 Intelligent, experience based monitoring

The role of the monitors is to pre-detect upcoming undesired operating conditions and faults and therefore to suggest maintenance actions. Each monitor holds exactly one target of detection. Attached with the monitor is an alarm message. A guideline may be attached, too, to support the operator to remedy the issue. Monitors do not require operator's interactions as they run autonomous and continuously observe a set of plant readings based on rule-based calculations/derivations. The monitors have a fixed sample rate, thus time based triggered by the monitor agent.

The CEXS monitors data and events of the control unit, the ALSPA P320 system. The ALSPA P320 system collects all events and measures from the power plant. The data is checked for feasibility and cleansed if necessary. The data and events are then analyzed, using experiences gained from past incidents to recognize upcoming errors or instable machine states as soon as possible. To achieve that, the incidents are stored in a historical data repository (HDR), to enable experts to evaluate new predictive monitors against past incidents. From these examinations the experts can derive a new set of analyzing rules and add them to the monitoring agent. Mechanisms and tools are provided with the CEXS, to analyze the incidents, to derive rules out of the upcoming experiences.

2.2 Interactive guidelines

In case of a fault or an operation stop, CEXS helps the operator to take the right steps. Therefore, after a fault/stop occurred, the system checks for existing advices. The matching advices are suggested to the operator as set of interactive guidelines from which the user can select the best fitting one. Independently from a fault/stop occurred, the operator is also able to search manually for an existing interactive guideline according to the semantic search principles presented in [Mo03]. The search for a solution is a diagnosis task which is covered in the ontology by appropriate problem solving rules. An interactive guideline advises the operator step by step about measurements to be taken and examinations to be done. Historical data for tests are available from the HDR which collects data from the ALSPA P320 and stores them over time. Finally the CEXS displays recommendations and solutions to the operator about the actions necessary to handle the fault or resume operation. The CEXS is capable for further extensions. New faults, actions or resumptions strategies may be recorded and stored within the corresponding interactive guideline.

2.3 Learning from Incidents

The CEXS interactive guidelines may be improved by lessons learned from past incidents. To be more precise CEXS supports the knowledge engineer (responsible for further improvement of the system) to recognize gaps and faults of the monitors and interactive guidelines and to improve them. Therefore CEXS stores a set of relevant information associated with each incident for further examinations.

CEXS provides a tool allowing the knowledge engineer to view/analyze the stored incidents and derive new experience from. The new experiences may lead to a set of new predictive rules for the monitor agent or to maintaining actions for the monitors or interactive guidelines. CEXS provides capabilities to be maintained by the technician.

All assessing of operating conditions associated with incidents is done by a knowledge engineer using the CEXS development tools. There is no “full automatic” assessment of operating conditions. The complexity and number of signals and measurements in the hydropower environment is huge, the number of incidents of the same type is low. Therefore automatic assessment without human intervening would fail.

3 Ontology

Subject of the monitoring of the CEXS initially are the turbine, the generator, and the unit transformers. For these aspects of the plant a set of monitoring rules and interactive guidelines are provided.

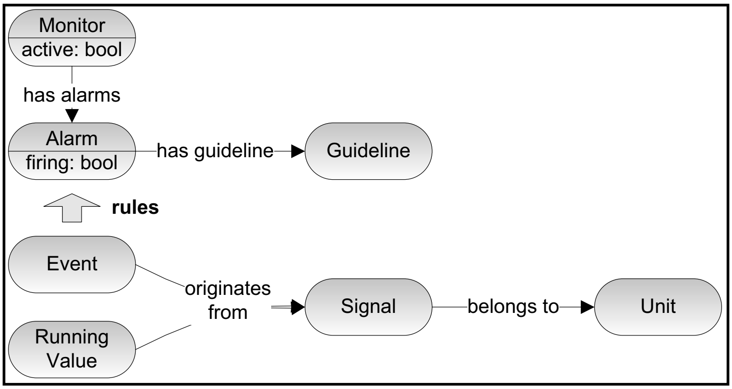


Figure 1: Excerpt from the ontology

3.1 Sensor based ontology for monitoring a power plant

The excerpt from the ontology in Figure 1 shows the central concepts of the ontology model. An instance of *Monitor* can be switched on or off (*activate* attribute). A monitor is assigned one or more alarms and for a monitor the time trigger frequency is defined. Monitors may be activated or deactivated. This information is again represented in the ontology (attribute *queryFrequency* and *isActive*). The alarms created by the monitors contain a warn message, the description of the alarm, a time period for which events have been considered for this alarm, and a flag indicating the *firing* of the alarm. A guideline can be attached to an alarm, guiding the operator then to handle the problem scenario defined by the alarm. The concept *Event* has a relation *originatesFrom* to *Signal*. *Running Value* holds the current value of *Signal*. This separate concept is necessary for computing reasons, as signals and events are only transferred over the bus system of the ALSPA P320 on value changes. A *Signal* itself originates from a certain unit, of which there are eight in the Bakun Hydroelectric Project.

The central concept is *Event*. Each event has a [time; value] tuple representing the occurrence time stamp and the corresponding value of the sensor. Complex relationships in this domain are described via F-logic rules [KLW95]. For instance, given the matter of affairs when an oil pump runs stable:

- The oil pump has a stabilization time of 5 minutes after the starting procedure.
- The start of the pump is indicated by signal value 1 from signal CL105

This could be represented in a rule like: “if the signal CL105 has the value *1* and the time stamp *T* then the pump runs stable at time *T+5* min. In F-logic such a rule is represented as follows (cf. also Figure 2):

```
?P[runsStable] ← ?E:Event[hasValue->1,
                          hasId-> CL105,
                         hasTimeStamp->?T,
                         originatesFromSensor->?S]
        and groovy_sbbf(“Duration”, ?currentTime:Timestamp, ?T, ?D)
        and?D > 5.0
        and ?S[belongsToUnit->?P].
```

In the modelling environment OntoStudio™ [An07] a rule editor is available which allows a rule being developed in an interactive graphical way. The rule is shown in Figure 2 in the graphical rule editor.

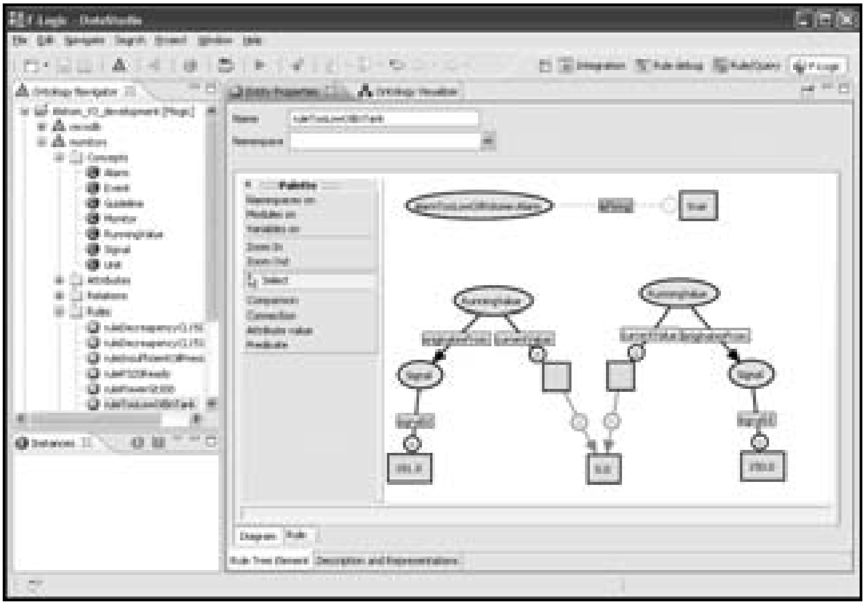


Figure 2: Graphical rule editor

The concrete sensor values are stored in a database. OntoStudio™ provides a comfortable mapping tool to map external data to the ontology via a mapping tool. Figure 3 shows such a mapping.

The schema of the database is shown on the left hand side providing tables like *EVENT*, *RUNNING_VALUE*, etc. On the right hand side our domain ontology is shown. The arrows show that *EVENT* is mapped to *SensorEvent* and the column *EVENT_ID* is mapped to the attribute *eventId* of concept *SensorEvent*. These mappings are evaluated during query time. So if a query is posed to the ontology like “give me all firing alarms of all active monitors” the sensor values are accessed during the evaluation of the query.

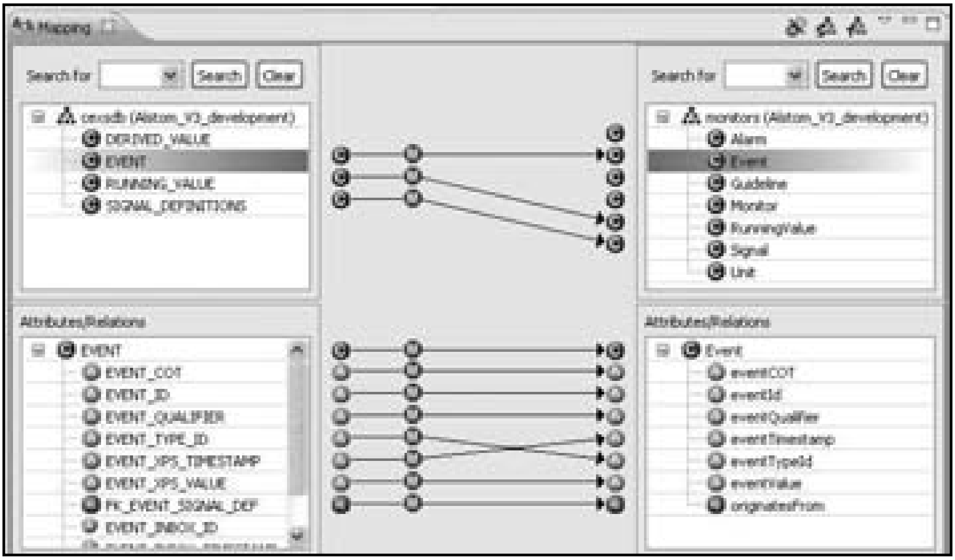


Figure 3: Mapping sensor data to the ontology

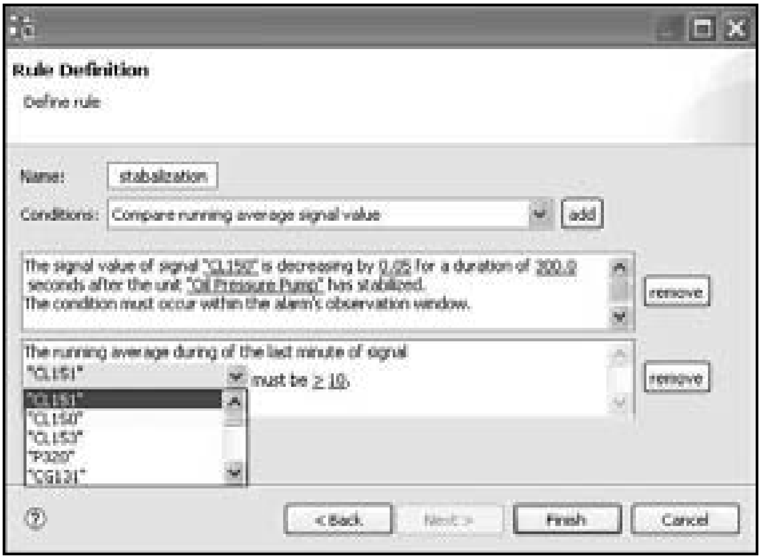
3.2 Monitors

A monitor continuously retrieves and analyzes actual data and as soon as it detects an issue or a critical upcoming problem it informs people or other systems about that. In our case the active monitors analyze the data from over 2,000 sensors per unit delivered by the power plant analyze them and give an alarm if upcoming problems are predicted.

The analysis of the sensor data together with their history may become very complex. Therefore powerful formalisms must be used to represent these methods. In our case again F-logic rules are used to describe monitors. These rules describe logical relationships between the components, the sensor data, and their history. A simple example for the monitor “Insufficient oil pressure of OPU“ could be: “if the signal with signal id 4711 has occurred and the sensor value for this signal has increased by more than 10% the last 10 minutes then the oil pressure is too low”.

Monitors are developed, maintained and deployed in the ontology development environment OntoStudio™. OntoStudio™ has been extended by a component for the development and administration of such monitors.

The alarm rules of the monitors have mostly the same pattern. They all have some conditions and they throw alarms, which equals to the same header structure. A specialized editor (rule wizard) has been developed for easier definition of these monitors. This editor follows the concept of the rule wizard in Microsoft Outlook (cf. Figure 4). Different templates show the rule in natural language. A click on a red part shows a selection list of instances of a concept which allows filling the concept template by an item in the selection list (instance of the concept). This rule editor automatically creates F-logic rules.



Rule Definition

Define rule

Name: stabalization

Conditions: Compare running average signal value

The signal value of signal "CL150" is decreasing by 0.05 for a duration of 200.0 seconds after the unit "Oil Pressure Pump" has stabilized.
The condition must occur within the alarm's observation window.

The running average during of the last minute of signal "CL151" must be ≥ 10.

"CL151"
"CL150"
"CL153"
"P320"
"CG121"

Figure 4: Rule wizard for monitor rules

Sometimes the computations for monitors are very complex and very procedural, which makes it very inconvenient and complex to represent these computations in a logical language like F-logic. The inference engine OntoBroker™ [De99] which evaluates these rules allows procedural calculations to be attached within Java built-ins. OntoBroker™ already contains a huge set of built-ins like basic mathematical operations, string operations, etc. The CEXS system required very complex mathematical operations beyond the available built-ins. For that purpose Mathematica has been attached to OntoBroker™ via its connector framework. Additionally the possibility for the development of domain specific built-ins in a procedural language was required. To cover this requirement a script editor in OntoStudio™ enables the development of built-ins in the dynamic Java scripting language Groovy.

These built-ins are hot-pluggable, which means they can be edited, adapted to current changes, tested in OntoStudio™ and immediately be deployed to the running inference system OntoBroker™ in the productive environment. Built-ins are used in F-logic rules as built-in predicates. In our example the built-in is evoked in the following way in a rule body: ... ← ... *groocy_sbbf("Duration", ?startTime, ?endTime, ?Period) ...*

3.3 Interactive guidelines and problem solving methods

To fulfill the interactive guideline requirements and guide the operator through complex faults and thus to resume operation, we have used our product SemanticGuide™.

The problem solving knowledge in SemanticGuide™ is designed generically and thus adaptable to any domain. In SemanticGuide™ we have included different problem solving methods: We use a simplified variant of the problem solving method Cover and Differentiate [EM99], because it was most suitable for the existing problem and requirements. Cover and Differentiate is the problem solving method, which was used in the expert system “Mole” [EDM86]. Cover and Differentiate is suitable for covering diagnosis problems, whose solution is a subset from a certain quantity of pre-defined solutions.

Secondly the user is able to use a decision tableau whose basic idea is to attach differentiating attributes to the solutions of a problem which itself represent a symptom. Based on these differentiating attributes, the rules within SemanticGuide™ decide, which symptom to observe next by excluding solutions having an attribute attached contrary to the answered one. Thus step by step the number of presented solutions is reduced.

Thirdly the user can choose to model the decision logic graphically as fault trees to describe problems and solutions (cause, actions, attachments and remarks).

For the CEXS project a combination of the second and third variant has been chosen.

The problem solving method was modelled itself as ontology. By using this approach, we represent a search algorithm for this diagnostic problem in a directed acyclic graph. The nodes represent the actual symptoms the user is examining, whereas the edges hold the paths, i.e. the answers the users can give.

4 Architecture

An overview of the interplay of the different mentioned components is given in Figure 5.

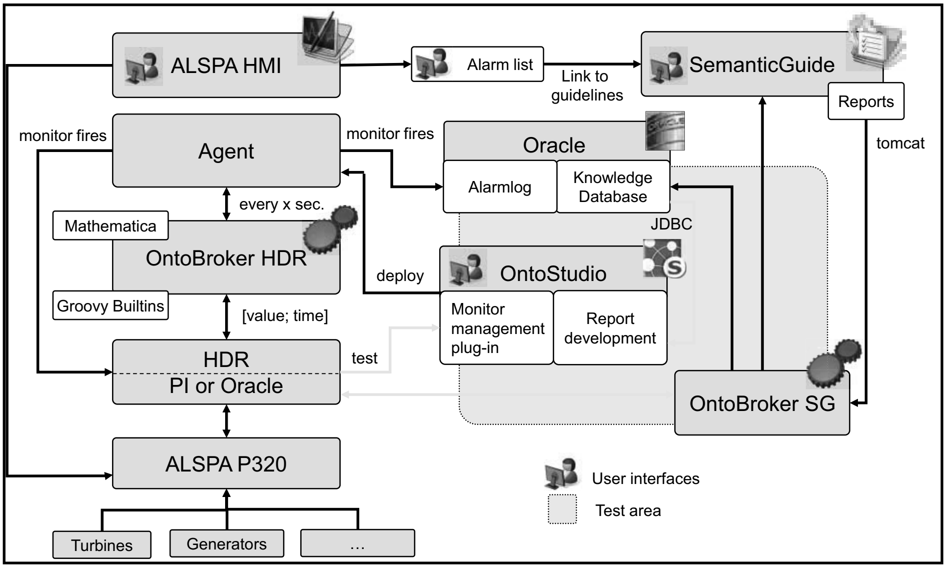


Figure 5: System architecture

The sensors in the different units of the power plant like turbines, cooling units, generators, pumps deliver their events and their signal values into the historical data repository (HDR) where they are stored for a longer time period to enable the agents to learn from the incidents. In the development environment OntoStudio™ the ontology, the rules and the monitors are developed, tested and maintained. OntoStudio™ integrates the reasoning engine OntoBroker™ for testing and debugging purposes. OntoStudio™ has access to the HDR to allow for testing rules and monitors on historical data. The HDR also lasts the current values and events in the *RunningValue* (c.f. chapter 0).

OntoStudio™ deploys the tested models to the productive runtime system CEXS. To keep process and workflow activities away from the inference engine OntoBroker™, we have implemented a semantic agent in CEXS based on the idea of [Be01]. This semantic agent acts for the user and manages

- the versioning of the ontology model with the means of the Content Repository API for Java (JCR) based on the WebDAV protocol
- the administration of the monitors and the scheduling of the queries to OntoBroker™
- the remote access from OntoStudio™ by means of the Java Management Extensions (JMX).

The semantic agent poses queries to the reasoning engine in the defined frequency and delivers alarms to the operator's ALSPA HMI (human machine interface) by invoking new alarm events on the ALSPA P320 bus system. The ALSPA HMI has been extended to be able to link to the advisory system which immediately creates proposals for solutions of the detected problems.

5. Conclusion

In this article we presented an application for monitoring sensor data of a power plant and to provide interactive guidelines to advise the operator to take the right steps. It provides mechanisms to learn from past incidents. Our monitors are based on semantic models, i.e. ontologies. Crucial is the flexibility to modify and maintain these models on the fly by the operators. Ontologies are very well suited for this task. They provide complex relationships in an understandable way on an abstract level. They are easy to change and easy to extend and immediately operational. Rules capture complex relationships among many factors and they define the monitors. OntoStudio™ the ontology modelling environment has been extended by administration parts for the monitors, for accessing historical sensor data, for deploying and versioning of ontologies and for easier creation and handling of rules. This application may be classified as a predictive analytics application. Predictive analytics is a sub area of business intelligence. Thus this paper shows also the benefits of semantic technologies in these fields.

6. References

[An07] Angele, J. et. al: Ontology Based Knowledge Management in Automotive Engineering Scenarios. In: Proceedings of ESTC 2007, 1st European Semantic Technology Conference, 31.05.-01.06.2007, Vienna, Austria

[Be01] Berners-Lee, T. et. al. The Semantic Web. Scientific American Vol. 284, Nr. 5 (2001). S. 34-43.

[De99] Decker, S. et. al. Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In R. Meersman et al., editor, Database Semantics: Semantic Issues in Multimedia Systems. Kluwer Academic, 1999.

[EDM86]Eshelman, L; McDermott, J.: MOLE: A Knowledge Acquisition Tool That Uses Its Head. In Boose, J.H.: Gaines, B. (Eds.): AAAI-Workshop: Knowledge-Acquisition for Knowledge Based Systems, Banff, Kanada, 1986.

[EM99] Eshelman, L: Mole: A Knowledge-Acquisition Tool for Cover-and-Differentiate Systems. In [Marcus 1988], 1988, p. 37-80.

[KLW95]Kifer, M.; Lausen, G.; Wu, J.: Logical Foundations of Object-Oriented and Frame-Based Languages. Journal of the ACM, 42:741{843, 1995.

[Mo03] Mönch, E. et. al: SemanticMiner - Ontology-Based Knowledge Retrieval. In: Special Issue of selected papers of the WM2003 in the Journal of Universal Computer Science (J.UCS).

[WP09] Predictive analytics. (2009, January 3). In Wikipedia, The Free Encyclopedia. Retrieved 12:05, January 12, 2009, from http://en.wikipedia.org/w/index.php?title= Predictive_analytics&oldid=261703565