Semantic Rules for Siemens Turbines

Gulnar Mehdi[1,2] Evgeny Kharlamov[3] Ognjen Savković[4] Guohui Xiao[4] E. Kalaycı[4]
Sebastian Brandt[2] Ian Horrocks[3] Mikhail Roshchin[2] Thomas Runkler[1,2]

[1]Technical University of Munich, Germany [2]Siemens AG CT, Munich Germany
[3]University of Oxford, UK [4]Free University of Bozen-Bolzano, Italy

1 Introduction

Motivation. Diagnostic systems play an important role in industry since they help to maximise equipment's up-time and minimise its maintenance and operating costs [16]. In the energy sector companies like Siemens often rely on *rule-based* diagnostics to analyse power generating equipment by, e.g., testing newly deployed electricity generating gas turbines [11], or checking vibration instrumentation [13], performance degradation [14], and faults in operating turbines. For this purpose diagnostic engineers create and use complex diagnostic rule-sets to detect equipment abnormalities.

An important class of rules that are commonly used in Siemens are *signal processing rules* (SPRs) that allow one *(i)* filter, aggregate, combine, and compare *signals*[1] coming from sensors installed in equipment and *(ii)* send notification messages when a certain pattern in signals is detected. *Authoring* SPR based rule-sets is challenging. We now discuss this challenge in details and then present our solution to address them.

Challenges with Authoring SPRs. The main challenge for authoring is that SPRs in most modern industrial diagnostic systems including the ones used in Siemens are highly *data dependent* in the sense that specific characteristic of individual sensors and pieces of equipment are explicitly encoded in SPRs. As the result for a typical turbine diagnostic task engineers have to write from dozens to hundreds of SPRs that involve hundreds of sensor ids, component codes, sensor and threshold values as well as equipment configuration and design data. E.g., a typical Siemens gas turbine has about 2,000 sensors and a typical diagnostic task is to verify that the purging[2] has ended; for the main flame component of a given turbine, this task requires around 300 SPRs, most of which are similar in structure but different in equipment specific data values.

Thus, there is a need in industry, and in particular in Siemens for a higher level diagnostic rule language that allows to express *what* the diagnostic task should do rather than *how* it should do it for specific equipment. Such language should be high level, data independent, while powerful enough to express in a concise way most of typical diagnostic tasks in Siemens.

Our Solution. We rely on *semantic technologies* to address the the above mentioned challenges. In particular we rely on *ontologies* [1] to define a novel SPR language and on *reasoning* [3] over ontologies to foster execution and maintenance of diagnostic tasks. In short, an ontology is a formal conceptualisation of the domain of interest that consists of a *vocabulary*, i.e., names of classes, attributes and binary relations, and *axioms* over the terms from the vocabulary that, e.g., assign attributes of classes, define relationships between classes, compose classes, class hierarchies, etc. Since ontologies

[1] Signals are are time stamped sequences of measurement values.

[2] Purging is the process of flushing out liquid fuel nozzles or other parts which may contain undesirable residues.

are specified using a formal logical language such as the W3C standardised ontology web language OWL 2, one can query ontologies and check their properties using reasoning that typically corresponds to logical entailment and implemented in many efficient state-of-the-art reasoning systems such as HermiT [15]. We refer the reader to [1] for more details on ontologies and reasoning.

In order to address the authoring challenge we propose:

- an SPR language *sigRL* that treats signals as first class citizens and allows signals to be processed (filtered, aggregated, combined, and compared) in a high level, declarative, and data independent fashion;
- semantic diagnostic programs that combine *sigRL* rules with diagnostic background knowledge captured using ontologies and allow users to express complex diagnostic tasks in an abstract fashion by exploiting both ontological vocabulary and queries over ontologies to identify relevant information (such as sensor ids and threshold values) about the equipment that should undergo the diagnostics.

Note that we designed *sigRL* in such a way that, on the one hand, it captures the main signal processing features required by Siemens turbine diagnostic engineers and, on the other hand allows for efficient execution of diagnostic programs.

2 Our Diagnostic Solution

We first illustrate *sigRL* with an example and then describe our system *SemDia*.

***sigRL* Diagnostic Language.** Consider a purging diagnostic task:

Verify that the purging ended in the main flame component of the turbine T1.

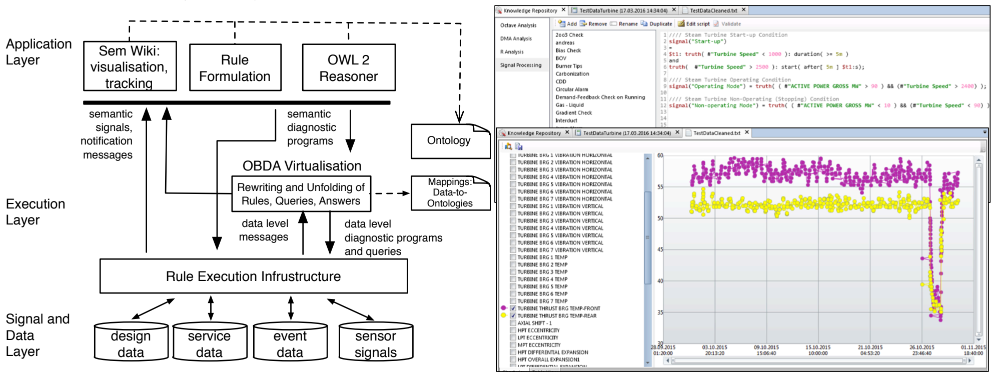
Intuitively this task requires to check in the turbine T1 that: *(i)* the main flame was on for at least 10s and then stopped, *(ii)* 15s after this, the purging of rotors in the starting-component of T1 started, *(iii)* 20s after this, the purging stopped. The fact that the purging of a rotor started or ended can be detected by analysing its speed, i.e., by comparing the average speed of its speed sensors with purging thresholds that are specific for individual rotors. The purging diagnostic program in our language *sigRL* can then consist of an ontology with one axiom: $\mathsf{SubClassOf}(\mathsf{RotorSensor}\ \ \mathsf{SpeedSensor})$. two signal processing expressions and one message rule:

$$\begin{aligned}
\mathsf{PurgingStart} &= \mathit{avg}\ \mathsf{rotorStart} : \mathsf{value}(>, \mathit{purgingSpeed}),\\
\mathsf{PurgingStop} &= \mathit{avg}\ \mathsf{rotorStart} : \mathsf{value}(<, \mathit{nonPurgingSpeed}),\\
\mathsf{msg}(\text{“}\mathit{Purging\ over}\text{”}) &= \mathsf{FlameSensor} : \mathsf{duration}(>, 10s) :\\
&\qquad \mathit{after}[15s]\ \mathsf{PurgingStart} : \mathit{after}[20s]\ \mathsf{PurgingStop}
\end{aligned}$$

For the complete description of *sigRL* we refer to our papers [10,7,9]. Here, the ontology defines a vocabulary over which one can write diagnostic rules in *sigRL*.

***SemDia* Diagnostic System.** The main functionality of our semantic rule-based diagnostics system *SemDia* is to author and maintain *sigRL* diagnostic programs, to deploy them in turbines, to execute the programs, and to visualise the results of the execution. We now give details of our system by following its architecture in Figure 1 (left) where the solid arrows indicate data flow and dashed arrows indicate—access to ontologies and mappings. There are three essential layers in the architecture: application, rule execution, and signal and data layers. Our system is mostly implemented in Java. We now discuss the system layer by layer.

Fig. 1. Left: Architecture of *SemDia*. Right (Screenshots): SPR editor (top), Diagnostic visualisation monitor (bottom)



Application Layer. On this layer, the system allows engineers to author, store, and load diagnostic programs by formulating sets of SPRs as well as message rules in *sigRL* and sensor retrieving queries. Such formulation is guided by the domain ontology stored in the system. In Figure 1 (right, top) one can observe a screenshot of the diagnostic program editor which is embedded in the Siemens analytical tool-kit. Another front end component is the semantic wiki that allows among other features to visualize signals and messages (triggered by programs), and to track deployment of programs in equipment. In Figure 1 (right, bottom) one can see visualisation of signals from two components of one turbine. The back end of the application layer relies on HermiT [15] ontology reasoning. Diagnostic programs formulated in the application layer are converted into XML-based specifications and sent to the rule execution layer that returns back messages and signals. We rely on the REST API to communicate between the application and execution layer of our system and the OWL API to deal with ontologies.

Execution Layer. On this layer we support semantic signals that are either *native*, that is, represented in terms of the diagnostic ontology as RDF triple, or *virtual*, that is obtained through the Ontology Based Data Access (OBDA) [12] component of *SemDia*. This component allows to present signals stored in relational databases as if they were native semantic. This requires to connect the relational signals to an ontology via declarative mappings. For the OBDA layer we rely on the extension of the Ontop system [2] developed during the Optique project [8,6,4,5] that takes care of transforming diagnostic programs written in *sigRL* into either SPRs written in the Siemens data-driven rule language or SQL. This transformation has two steps: rewriting of programs and queries with the help of ontologies (at this step both programs and queries are enriched with the implicit information from the ontology), and then unfolding them with the help of mappings. Moreover, the execution layer takes care of planning and executing rules and queries received either from the rule management or OBDA component. If the received rules are in the Siemens SPR language then the rule executor instantiates them with concrete sensors extracted with queries and passes them to the Drools Fusion the engine used by Siemens. If the received rules are in SQL then it plans the execution order and executes them together with the other queries.

Signal and Data Layer. On this layer we store all the relevant data: turbine design specifications, historical information about services that were performed over the turbines, previously detected events, and the raw sensor signals. Currently *SemDia* support PostgresQL, Teradata, as well as Sparksee.

3 Demonstration Overview

Demo attendees will be able to learn how to do diagnostics of Siemens turbines with *sigRL* diagnostic programs. To this end we prepared a deployment of our *SemDia* system on data from 50 Siemens power generating turbines, a diagnostic ontology, and a catalogue of 15 diagnostic tasks. The attendees will be able to load preconfigured diagnostic programs, deploy and execute them, author their own diagnostic programs, and try out our provenance computation and program verification services.

Acknowledgments This research is supported: EPSRC projects MaSI^3, DBOnto, ED^3; and the Free University of Bozen-Bolzano project QUEST.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
2. Calvanese, D., Cogrel, B., Komla-Ebri, S., Kontchakov, R., Lanti, D., Rezk, M., Rodriguez-Muro, M., Xiao, G.: Ontop: Answering SPARQL queries over relational databases. Semantic Web 8 (2017)
3. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. Journal of Automated Reasoning 39(3) (2007)
4. Horrocks, I., Giese, M., Kharlamov, E., Waaler, A.: Using semantic technology to tame the data variety challenge. IEEE Internet Computing 20(6), 62–66 (2016)
5. Kharlamov, E., Brandt, S., Jiménez-Ruiz, E., Kotidis, Y., Lamparter, S., Mailis, T., Neuenstadt, C., Özçep, Ö.L., Pinkel, C., Svingos, C., Zheleznyakov, D., Horrocks, I., Ioannidis, Y.E., Möller, R.: Ontology-Based Integration of Streaming and Static Relational Data with Optique. In: SIGMOD (2016)
6. Kharlamov, E., Jiménez-Ruiz, E., Zheleznyakov, D., Bilidas, D., Giese, M., Haase, P., Horrocks, I., Kllapi, H., Koubarakis, M., Özçep, Ö.L., Rodriguez-Muro, M., Rosati, R., Schmidt, M., Schlatte, R., Soylu, A., Waaler, A.: Optique: Towards OBDA systems for industry. In: ESWC Satellite Events. pp. 125–140 (2013)
7. Kharlamov, E., Savkovic, O., Xiao, G., Penaloza, R., Mehdi, G., Horrocks, I., Roshchin, M.: Semantic rules for machine diagnostics: Execution and management. In: CIKM (2017)
8. Kharlamov, E., Solomakhina, N., Özçep, Ö.L., Zheleznyakov, D., Hubauer, T., Lamparter, S., Roshchin, M., Soylu, A., Watson, S.: How semantic technologies can enhance data access at siemens energy. In: ISWC. pp. 601–619 (2014)
9. Mehdi, G., Kharlamov, E., Savkovic, O., Xiao, G., Kalayci, E.G., Brandt, S., Horrocks, I., Roshchin, M., Runkler, T.: Semdia: Semantic rule-based equipment diagnostics tool. In: CIKM (2017)
10. Mehdi, G., Kharlamov, E., Savkovic, O., Xiao, G., Kalayci, E.G., Brandt, S., Horrocks, I., Roshchin, M., Runkler, T.: Semantic rule-based equipment diagnostics. In: ISWC (2017)
11. Mitchell, J.S.: An introduction to machinery analysis and monitoring. Pennwell Books (1993)
12. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. J. Data Semantics 10, 133–173 (2008)
13. Randall, R.B.: Vibration-based condition monitoring: industrial, aerospace and automotive applications. Wiley (2011)
14. Rao, B.K.N.: Handbook of condition monitoring. Elsevier (1996)
15. Shearer, R., Motik, B., Horrocks, I.: HermiT: A Highly-Efficient OWL Reasoner. In: OWLED (2008)
16. Vachtsevanos, G., Lewis, F.L., Roemer, M., Hess, A., Wu, B.: Intelligent Fault Diagnosis and Prognosis for Engineering Systems. Wiley (2006)