ICM-Wind: Semantics-Empowered Fluid Condition Monitoring of Wind Turbines

Matthias Klusch, Ankush Meshram, Patrick Kapahnke
German Research Center for Artificial Intelligence (DFKI)
Saarbruecken, Germany

Andreas Schuetze
Centre for Mechatronics and Automation Technology (ZeMA)
Saarbruecken, Germany

ABSTRACT

We present the first system, called ICM-Wind, for semantics-empowered fluid condition monitoring (FCM) in wind turbines. It monitors the condition of fluids in the wind turbine gearbox, recognizes actual and the onset of failures of FCM sensors and components installed on the gearbox, and provides knowledge-based failure diagnosis support to non-experts. For this purpose, the ICM-Wind system performs semantic sensor data analysis by applying semantic technologies for interpreting the state of turbine parts and answering questions related to their maintenance. Domain knowledge is encoded in OWL2 and with SPIN rules. Fault detection and diagnosis queries are answered by use of the semantic reasoners Fact++, STAR, TopSPIN rule engine, and SwiftOWLIM store. The system prototype was successfully tested in cooperation with the HYDAC Filter Systems GmbH based on given selected samples of a two-year recording of FCM multi-sensor and operational data for two wind turbines of a regional on-shore wind farm operated by the ABO Wind AG.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Knowledge Representation-Semantic Networks

General Terms

Algorithms

Keywords

Condition monitoring of wind turbines, semantic sensor data, semantic reasoning

1. INTRODUCTION

Today, the major strategy of operators of on-shore or off-shore wind farms to avoid the very costly and catastrophic failures or breakdowns of a wind turbine is to maintain its critical components based on their actual condition, rather than by a fixed scheduled, preventive replacement or mere reactive maintenance. Condition-based on-site or remote maintenance requires condition monitoring (CM) which encompasses continuous data collection, fault recognition and fault diagnosis [10]. The conventional wind turbine drive train consists of a rotor, mainshaft and bearing, gearbox with main turbine gear, and power generator, all mounted on a common bedplate in the turbine nascell on top of a turbine tower. Within the complex wind turbine system, the main gear which transfers the rotation from the rotor to the generator is a critical component. To ensure its reliable operation, the condition of the gear is determined by current condition monitoring (CM) systems through the continuous monitoring of its vibrations or the quality of lubricating fluids to recognize the onset of wear and faults of the gear based on complex sensor and signal processing [15, 5, 9]. In many wind turbines, a specific fluid condition monitoring (FCM) system is integrated with the gearbox which filters the oil continuously and cools it at higher operating temperatures to ensure consistent lubrication of the gear and prevent fast degradation of the oil. Such FCM systems are in particular equipped with multiple, networked sensors for monitoring various relevant physical and fluid parameters like gear speed, temperature, pressure, metallic and particle contamination of the lube oil.

However, current approaches to fluid condition monitoring of wind turbines still require a human engineer with extensive domain expertise to manually interpret the highly complex interdependencies between measured sensor and operational data and various system conditions for failure recognition and diagnosis. The main challenge of intelligent FCM is to predict the remaining oil filter lifetime, to identify sensor and operational data patterns that indicate the onset of failures, and to provide knowledge-based failure diagnosis support to non-experts. This goes far beyond currently available FCM systems which only perform fault detection mainly based on multi-variate statistical sensor data analysis [17]. On the other hand, only few approaches to intelligent CM exist which employ means of AI like neural networks [1, 11], causal graphs and model-based reasoning [14], agents [12] and semantic reasoning [7] for fault detection and diagnosis, though none are available for intelligent FCM of wind turbines yet. Thus we developed the first system for intelligent FCM of wind turbines, named ICM-Wind, which uses in particular semantic technologies in order to address the aforementioned challenges. The ICM-Wind system integrates a specific FCM subsystem for wind turbine gearboxes developed by our customer HYDAC Filter Systems GmbH with new components for off-line statistical and semantic analysis of sensor and operational data in order to answer certain types of queries related to failure detection and diagnosis, and system information. The semantic modelling of the FCM domain in a specific domain ontology in OWL2 under OWL-Horst semantics, the SPIN rules for sensor fault detection, as well as the identification of important types of FCM-related analysis queries to be answered by the system in practice have been achieved in close collaboration with engineers and domain experts of HYDAC [4].

In this paper, we focus on the semantic analysis component of the ICM-Wind system which exploits the TopSPIN rules engine, the SwiftOWLIM store, and the semantic reasoners Fact++ and STAR [8], either individually or in combination, in order to answer given FCM-related analysis queries off-line as required. The experimental performance evaluation of this component for given samples from a two-year recorded FCM and operational data volume for two GE-1.5sl wind turbines of an on-shore wind farm operated by the ABO Wind AG [19] revealed that the results and average response times of its semantic query processing are reasonable. The ICM-Wind system is considered by HYDAC GmbH as innovative for their FCM product portfolio [20] and useful for its targeted application in practice.

The remainder of the paper is structured as follows. After an overview of the overall system architecture and its components in section 2, we describe the specific ICM-Wind domain ontology according to which the measured FCM sensor data is semantically encoded in Section 3. The FCM-related query processing by the semantic analysis component and its experimental performance evaluation are presented in sections 4 and 5. We dicuss related work in section 6 before we then conclude the paper.

2. ICM-WIND SYSTEM ARCHITECTURE

Requirements and architecture. The main ICM-Wind system requirements which were given to us by our customer HYDAC are as follows: The system bases on the HYDAC FCM system integrated with a wind turbine gearbox and is able to (a) predict the condition, especially the remaining useful operating time of its oil filter, (b) detect the onset of faults of sensors and the main turbine gear in the gearbox, and (c) provide knowledge-based support for fault diagnosis to the engineer. In particular, our customer required that the system is able to correctly answer a given set of different types of informal, high-level FCM-related analysis queries (cf. Sect. 4) over given 1-/3-/5-/7-day recordings of FCM sensor and operational data of a wind turbine gearbox. Further, it was required that the analysis results are provided not in real-time during the data recordings but off-line within a maximum of three hours for a 7-day recording in practice.

These requirements are satisfied by our ICM-Wind system which component-based architecture (cf. Figure 1) is shortly described in the following. In this paper, we focus on the system component for semantic sensor data analysis, and refer the interested reader for more details on the other system components to [4].

HYDAC FCM system. The HYDAC fluid condition monitoring subsystem of the ICM-Wind system is integrated

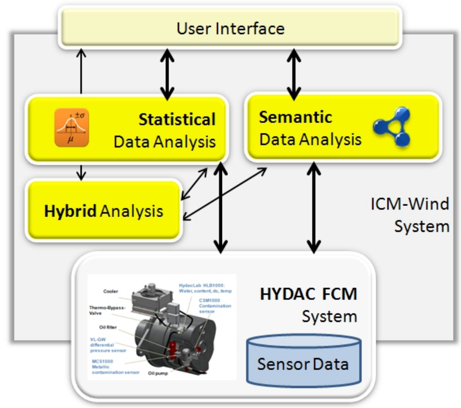

Figure 1: Components of the ICM-Wind system.

with a wind turbine gearbox. It consists of the oil filter, oil sump, oil pump, oil cooling loop with thermo-bypass valve of the gearbox, and the special set of 11 wired FCM sensors for monitoring physical parameters (absolute and differential pressure at the oil filter, temperature at different points in the oil loop, and rotational speed of the gear) and fluid, i.e. lube oil, condition parameters (dielectric constant, metallic and particle contamination). The collected multi-sensor data are preprocessed and stored in CSV files by the HYDAC-FCM system from which it is accessible by both the statistical and the semantic data analysis component via REST service operations but not yet via OGC sensor web standards [18].

Statistical analysis component. This component basically applies Linear Discriminant Analysis (LDA) to all recorded sensor and turbine operational data as well as secondary features like the increase and variation of measured differential pressure. It computes, in particular, a distinct trend of subsequent oil filter changes which also reflects the aging of the overall wind turbine gear in order to eventually classify the actual oil filter condition (new, medium, advanced aging), and to display deviations from the trend which indicate potentially harmful conditions of the turbine gear.

Semantic analysis component. This component performs the semantic sensor data encoding and analysis (cf. Figure 2). The sensor data provided by the HYDAC FCM system is RDF-encoded according to a specific domain ontology in OWL2 under OWL-Horst semantics [13] and the actual FCM system configuration data. This is followed by the evaluation of fault conditions of sensors with a set of SPIN rules (cf. Section 3).

The RDF-encoded and fault condition evaluated sensor data set is then transfered to the semantic data reasoning environment (SDRE) which consists of a query processor and functional plug-in modules for semantic query answering and reasoning. In its current version, the SDRE incorporates the non-commercial in-memory version of the RDF triple store SwiftOWLIM, and the semantic reasoners Fact++ and STAR [8]. The SwiftOWLIM store and the reasoner Fact++ are used to answer SPARQL, SPARQL-DL, respectively, DL queries over the domain ontology with a materialized fact base which each of these module creates internally from the given semantic sensor data set. The STAR reasoner cre-

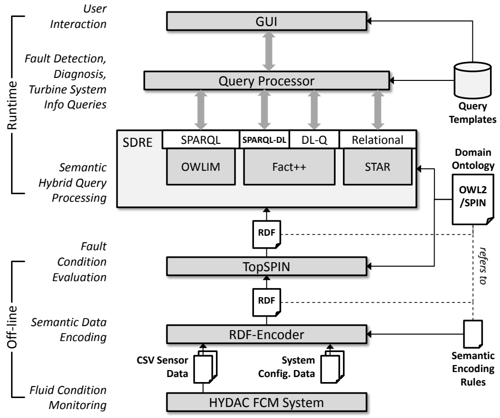

Figure 2: Architecture of the semantic analysis component.

ates an internal RDF graph representation of the ontology with a non-materialized fact base it then utilizes to answer RDF object-relational queries. Users of the system can enter different types of queries for fault detection, diagnosis and information by instantiating the respective query templates which are provided by the user interface. The query processor of the SDRE then processes these queries with either one or multiple of the SDRE modules in combination depending on the considered query type (cf. Section 4).

Hybrid analysis component. This system component is concerned with two types of queries whose answering requires the combined functionality of the statistical and semantic data analysis components. For example, the answering of the user query *"What is the actual condition of oil filter OF4?"* requires the semantic analysis component to check the result of the respective LDA analysis from the statistical component. If this result is not available in the fact base yet, the hybrid component automatically calls the statistical component to produce it, and then the semantic component to update its fact base and re-run the query processing. In turn, if the LDA analysis result of the statistical component for oil aging shows a faulty sensor signal for oil temperature (S_{Temp}), the hybrid component indicates this to the user and proposes the use of alternative sensors with a semantically equivalent parameter for the analysis. That is, it calls the semantic component to answer the query *"Which property P measured by which sensor can be used instead of the property S_{Temp}?"* and, in case of a positive result, then triggers a re-run of the LDA analysis by the statistical component with the sensor data for P instead of S_{Temp}.

3. DOMAIN MODELING AND SEMANTIC DATA ENCODING

We modeled the FCM domain for wind turbines in a specific domain ontology called ICM-Wind ontology in close collaboration with engineers and domain experts of our customer HYDAC. In the following we provide an overview of this ontology, its extension with fault detection rules, and the semantic encoding of sensor data according to this ontology.

ICM-Wind Ontology: Overview. The ICM-Wind ontology represents knowledge about the wind turbine domain in general and the FCM domain in particular. For this purpose, it aligns appropriate concepts and properties defined in the standard W3C-SSN ontology [3] for semantic sensor networks with specific ones we defined in compliance with (a) the ISO-13374 standard [6] for condition monitoring of machines, and (b) the given specification of the HYDAC FCM system for wind turbine gearboxes. Parts of the ICM-Wind ontology related to the wind turbine, FCM subsystem, sensors, and faults are shown in figures 3 and 4.

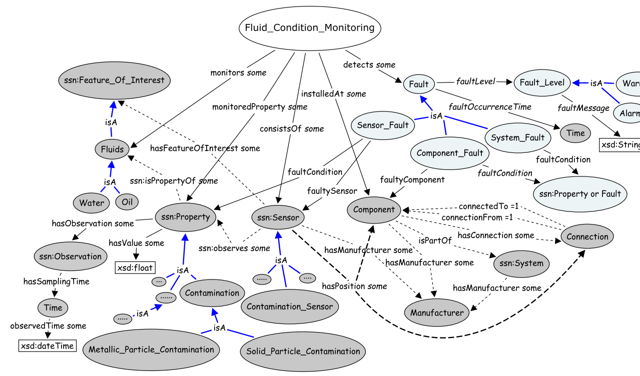

Figure 3: Part of ICM-Wind ontology representing FCM, sensor and fault concepts.

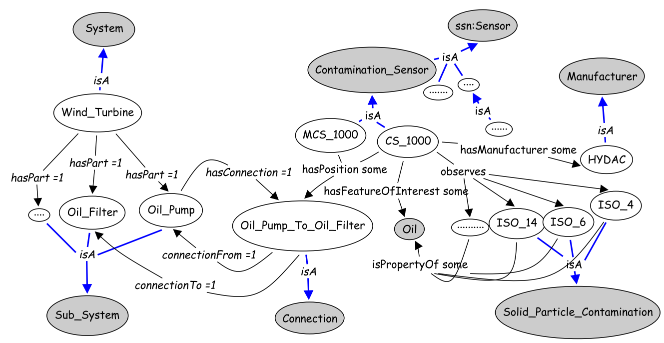

Figure 4: Part of ICM-Wind ontology representing wind turbine and HYDAC sensor concepts.

The static conceptual domain knowledge was modeled jointly with domain experts from HYDAC and specified in the standard ontology language OWL2, more concrete, in the fully RDFS compatible OWL-Horst fragment of OWL2 [13] (a.k.a. OWL-Tiny) which was expressive enough for our modeling of the considered domain. This OWL fragment is efficiently implemented in the RDF triple store SwiftOWLIM as well as supported by the reasoners Fact++ and STAR of the semantic analysis component. In its initial version, the concept base (TBox) of the ICM-Wind ontology consists of 111 concepts, 28 relations (properties) and 4 XSD data types (float, string, bool, dateTime) in total, and is provided by the semantic analysis component to its RDF-encoder module for the semantic encoding of sensor data as well as to its TopSPIN rule engine and SDRE modules for further semantic data analysis (cf. Sect 4).

Fault detection rules. The ICM-Wind ontology is extended with a set of 24 SPARQL-SPIN rules for the detection of (a) functional faults of sensors, and (b) property value-based failure conditions of the wind turbine system. In particular, each fault detection rule is uniquely associated with a sensor, component or property class as an extension of its definition in the ontology. Sensor fault detection rules determine for each property observed by the specific sensor whether its measured data are within a ISO standard-based value range, hence indicate whether the sensor is malfunctioning or not. The second type of rules is checking whether the measured property values within the specified ranges are tolerable or critical with regard to the monitored condition of the wind turbine gearbox. In our case, the respective limits of FCM-related sensor parameter values for certain failure conditions were given to us by the domain experts from HYDAC. Though SPARQL-SPIN is not a standard, we choose it for reasons of customization of user-defined mathematical functions which can be employed to specify failure conditions in SPARQL queries which goes beyond the expressivity of SPARQL and cannot be evaluated by any of the SDRE modules but the TopSPIN rule engine for a pre-detection of faults (cf. Sect 4.2). A simple example of the semantic representation of a contamination sensor with its assigned set of SPARQL-SPIN rules for sensor and property value-based system fault detection is shown in figure 5 (denoted in an abstracted form).

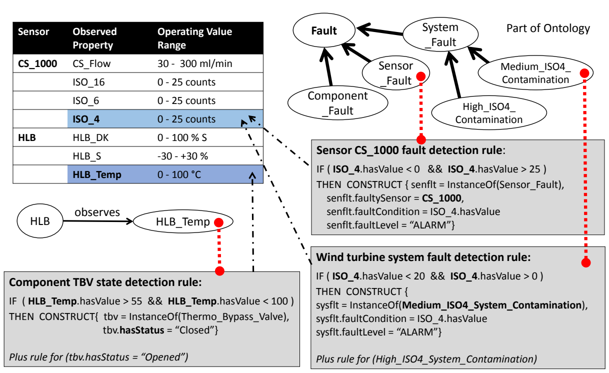

Figure 5: Fault and state detection rule examples

Semantic encoding of sensor data. The semantics of measured sensor data is encoded in compliance with concepts, relations and data types defined in the ICM-Wind ontology. For this purpose, the recorded raw sensor data is cleaned, processed and formatted in a tabular CSV format by the HYDAC FCM system which then transfers the CSV sensor data to the RDF-encoder module for its semantic encoding. Each observation record for each sensor in this data set is transformed by the RDF-encoder to a corresponding set of assertional RDF statements on property values of an object of the respective sensor concept defined in the domain ontology. For this purpose, the labels of sensors and measured properties as well as data types in the CSV data set are uniquely mapped to the corresponding names of concepts in the ontology. This yields the RDF-encoded semantic sensor data set for the given sensor data recording which is then transfered by the RDF-encoder to the TopSPIN rule engine for its pre-detection of faults, and further on to the SDRE for answering the given set of analysis queries for system information, fault detection and diagnosis.

4. SEMANTIC DATA ANALYSIS

The ICM-Wind system can answer selected types of FCM related analysis queries in the wind turbine domain which were considered important by the engineers at HYDAC Filter Systems GmbH in practice. In this paper, we focus on the processing of only few but representative examples of these given queries for system information, fault detection and diagnosis by the semantic data analysis component (cf. figure 2) of the ICM-Wind system.

4.1 System Information

The semantic analysis component of the ICM-Wind system utilizes the SDRE modules to provide information about the quantity, position and functionality of, as well as the semantic relations between components and sensors of the wind turbine according to the domain ontology. For this purpose, the system provides the user with a set of specific query templates for entering a query. Depending on the selected type of query template the query processor rewrites the user query as a query in either SPARQL, or SPARQL-DL, DL, or STAR-Q which then is answered by SwiftOWLIM, Fact++, or STAR, respectively. For example, the informal user query *"What sensor types can measure air temperature?"* is rewritten as a SPARQL-DL query as follows:

```
SELECT ?sen
WHERE{ SubClassOf(?sen, Sensor).
 PropertyValue(?sen, measuresProperty, ?prop).
 SubClassOf(?prop, Temperature).
 PropertyValue(?sen, hasFeatureOfInterest, Air)
```

and answered by the SDRE with its Fact++ module over the materialized fact base of the domain ontology. Another example is the query *"Which sensor is most specifically similar to the oil temperature sensor?"* which is rewritten as a SPARQL-DL query as follows:

```
SELECT ?spec
WHERE DirectSubClassOf(?spec, wto:OilTempSens)
```

and, again, answered by the SDRE with Fact++ through its efficient classifying of the query concept (in the WHERE-clause) into the ontology and returning the set of direct logically subsumed concepts. Besides, the query *"Which types of sensor property measurements refer to oil temperature?"* is answered with Fact++ by its returning of the properties $\{MCS_{Temp}, HLB_{Temp}\}$ which it determined as to be semantically equivalent with the query concept. Similarly, queries like *"Which sensors measure oil temperature and oil contamination with metallic particles of size larger than 300 micronmeter?"* can be answered by the SDRE as well.
Likewise, the system can determine expensive sensor redundancy in a designed FCM system which is modeled in the given concept base by answering the respective DL queries for subsumption-based semantic equivalence (or overlapping) of sensor concepts with the semantic reasoner Fact++.
Finally, information queries which are concerned with the discovery of semantic relation between components and/or sensors are answered by the SDRE with its STAR reasoner. For example, the user query *"How are the sensors CSM and SpeedSensor related?"* is rewritten by the query processor as the STAR query STAR({CSM, SpeedSensor}). The answer to the user by the query processor is a textual statement that describes the (set) of shortest property paths between both sensor objects in the domain ontology. In fact, the STAR reasoner of the SDRE computes an approximated solution

of the corresponding NP-complete Steiner-Tree problem [8] based on an internal graph representation of the ontology with a non-materialized fact base.

4.2 Semantic-Based Fault Detection

The semantic-based detection of faults of sensors and monitored system components is performed in two phases: First, the TopSPIN rule engine determines sensor faults, component states, and certain system failure conditions based on the non-materialized RDF-encoded sensor data set. Second, this pre-detection of faults yields a respectively extended semantic sensor data set which finally serves the SDRE as a basis for its answering of fault detection queries given by the user.

Pre-detection of faults and states off-line. As mentioned above, the SPARQL-SPIN rules for sensor and system fault detection are included in the sensor concept definitions of the ontology, and are evaluated by the TopSPIN rule engine against the semantic sensor data set provided by the RDF-encoder. Whenever a sensor fault detection rule is firing, that is the failure condition specified in the WHERE-part (or rule body) of the instantiated SPARQL query is satisfied, the CONSTRUCT-part (or rule head) adds a new RDF-encoded object of type `Sensor Fault` to the data set. This sensor fault object (cf. fig. 3) describes the name of the malfunctioning sensor, the level and time of occurrence of the fault. Likewise, a detection rule for some system fault which is associated with one or multiple physical properties measured by one sensor is adding RDF-encoded information about property value-based system failure conditions to the sensor data set whenever this rule evaluates to true. Finally, rules for the detection of states of components like the thermo-bypass valve of the FCM system are adding RDF-encoded information on the state of the component such as whether the valve is opened or closed. Such state information is not included in the original sensor data. There are no component fault detection rules evaluated against the sensor data set in this pre-detection phase. In any case, the evaluation of these three types of detection rules does not require implicit knowledge to be inferred from the semantic sensor data set, hence the data set does not need to be materialized in prior. The RDF-encoded results of this pre-detection phase are added to the original semantic sensor data, and then passed on to the SDRE for further analysis.

Fault detection query answering with the SDRE. Representative example of fault detection queries are *"Is the thermo-bypass valve faulty?"*, *"Is the oil filter of the FCM system clogged?"*, and *"What are the components of the FCM system with critical faults?"*. The query processor utilizes the SDRE modules SwiftOWLIM and STAR but not Fact++ for answering these queries, and, in contrast to the pre-detection phase, requires the inference of implicit knowledge from the extended semantic sensor data set by SwiftOWLIM. For example, the state of the thermo bypass valve (TBV) of the oil cooling mechanism of the turbine gearbox is not directly measured by the FCM system, hence is not explicitly encoded in the semantic sensor data set by the RDF-encoder. Therefore the only way to determine its condition is to compare the oil temperatures before and after the cooler which are measured by some sensors: The TBV is malfunctioning if the temperature difference is greater than 10 degree Celsius, and the TBV is closed. Thus, the first fault detection query above is answered by the query processor by combining the results of its sequential processing of SPARQL and STAR queries: It finds those sensors which are positioned before and after the cooler by evaluating STAR queries ({?sensor1, Cooler, ?sensor2}) for all sensor objects in the materialized fact base of SwiftOWLIM with SPARQL. The result of the STAR reasoner are the statements "HLB hasPosition [connection1] connectedTo Cooler" and "ETS hasPosition [connection2] connectedFrom Cooler" which is interpreted by the query processor as the sensors HLB and ETS being positioned before, respectively, after the cooler. It uses these two sensor objects as binding arguments of the call of a SPARQL query which determines whether the difference of the temperatures which are measured by these sensors as their properties (HLB_{Temp}, T_{in}) exceeds the given limit within what time periods.

The fact that both temperatures are indeed oil temperatures was infered from the ontology during materialization of the fact base by SwiftOWLIM. Information about the TBV state which is required to check in the SPARQL query whether the TBV is closed during the respective time periods is already present in the non-materialized fact base, that is the extended sensor data set SwiftOWLIM received from the TopSPIN rule engine.

Finally, the CONSTRUCT-part of the SPARQL query is adding a new TBV object of the component fault concept along with the fault information as its properties to the fact base. Please note that the query processor uses result caching to avoid redundant query evaluations.

4.3 Semantic-Based Fault Diagnosis

In general, fault diagnosis aims at finding the reasons of fault occurrence. The semantic analysis component of the ICM-Wind system focuses on the answering of diagnosis queries which are concerned with (a) providing information about detected faults, as well as (b) the determination of relations between sensor, component and system faults which were detected within the same time period. For example, the user-given diagnosis query *"Why and when did the TBV fault occur?"* (cf. Sect 4.2) is rewritten by the query processor as a SPARQL query to be evaluated by SwiftOWLIM. As a result the condition under (critical temperature difference and closed TBV) and the time periods within which the fault occurred is returned to the user. Note that this information was added in the fault detection phase into the fact base.

As for an example of the second type of diagnosis queries, consider the user query *"How are the faults of the contamination sensor MCS and the oil pump which are detected within the same period related?"* In this case, the query processor first issues a SPARQL query to SwiftOWLIM to obtain a list of pairs of those sensor and component fault objects for the given MCS and oil pump which values for their time occurrence property are within the same time period. For each of these pairs the STAR reasoner is called to evaluate the STAR query STAR({?mcs-flt, ?oilpump-flt}) in order to determine the semantic relation between the fault objects. This yields a set of textual statements each of which describing the shortest property paths between the considered fault objects in the undirected RDF graph of the ontology with (non-materialized) fact base. For example, the computed path (`mcs_flt1 faultySensor mcs`)
→ (`mcs1 rdf:type MCS_1000`)
→ [`MCS_1000 hasConnection Oil_Pump_To_Oil_Filter`]
→ [`Oil_Pump_To_Oil_Filter connectedFrom Oil_Pump`]

→ (oilpump1 rdf:type Oil_Pump)
→ (oilpump_flt1 faultyComponent oilpump1)
is compared by the query processor with a set of generic path patterns like

$$\exists X, Y, C : (X \texttt{ hasPosition } Y, Y \texttt{ connectedFrom } C)$$

each of which is associated with a natural text template to be instantiated with the respective path elements. In the example, the overall answer to the user is *"The faulty sensor mcs1 is positioned after the faulty component oilpump1: The sensor fault mcs_flt1 does not affect the component fault oilpump_flt1, that occured within time period 05-12-2011:21:13, 05-12-2011:21:18."* The time period information was taken by the query processor from the result of the SPARQL query evaluation before.

5. EVALUATION

The semantic data analysis component of the ICM-Wind system has been evaluated for its performance in terms of average query response times for different FCM related analysis queries on given sets of 1-/3-/5-/7-day recordings of sensor data. The component is implemented as a client-server Java web application using Google web toolkit.

5.1 Evaluation Setting

The evaluation experiments were performed on a mass-market notebook with following configuration: Intel(R)Core(TM) i7-2600K CPU@3.40 GHz with 16.0 GB RAM, JDK 1.7 with 14 GB Max JVM Heap Space, and Windows 7 Enterprise Service Pack 1 OS. The SDRE in its current version consists of the SwiftOWLIM (owlim-lite 5.3) triple store, the DL reasoner FaCT++ 1.6, and a recent revision of the STAR reasoner, while the OWL-API 3.4 is used for interaction between the triple store and the DL reasoner.

Test data set. The test data for our performance tests are two-year recorded and pre-processed FCM multi-sensor and operational data volumes for two GE-1.5sl wind turbines of an on-shore wind farm operated by the ABO Wind AG which were provided to us in CSV format by FCM experts of HYDAC Filter Systems GmbH. The fault detection was tested over increasing sizes of data recordings within one week selected by the experts from the overall test data set with recording periods of one day (1440 observations), three days (4320 observations), five days (7200 observations), and seven days (10080 observations). The given weekly test data set reflected, for example, a thermo-bypass valve fault and an oil filter blockage occurred on the third, respectively, fifth day which were correctly detected and diagnosed by the analysis component.

Performance measures. The performance of the semantic encoding process was evaluated in terms of the total encoding time which is the sum of time needed to (a) RDF-encode the raw sensor data and (b) to generate the extension of this data set by the TopSPIN rule engine during the fault pre-detection phase. The SDRE data loading time is concerned with measuring the time needed by its modules to be ready for query answering whenever the SDRE is provided with a new semantic sensor data set; this loading time includes the time needed by (a) the triple store to materialize the data set, (b) the DL reasoner Fact++ to prepare the same and its internal pre-computations, and (c) the STAR reasoner to generate its internal graph representation of the non-materialized data set. We measured the query response times of the SDRE for different types of analysis queries over different sizes of test data volumes.

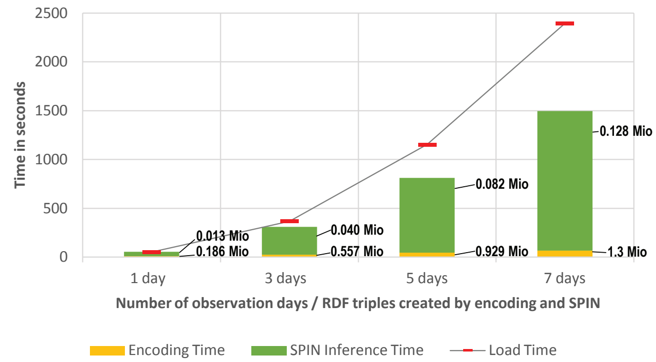

Figure 6: Encoding and loading times for FCM observation data of different sizes.

Test queries. From the exhaustive list of FCM-related analysis queries given by the expert a few representative user queries for system information, fault detection, and fault diagnosis to be performed by the analysis component with different kinds of semantic query answering and reasoning (SPARQL, SPARQL-DL, STAR, DL) were selected for preliminary performance evaluation: **Q1.** *"Are there any redundant sensors in the FCM system of the turbine gearbox? If yes, list them."*: SysInfo, **(DL)**; **Q2.** *"Which sensors measure oil temperature and oil contamination with metallic particles of size larger than 300 micronmeter?"*: SysInfo, **(SPARQL-DL)**; **Q3.** *"What faults are present in the turbine gearbox and which of these are critical?"*, Fault detection, **(SPARQL)**; **Q4.** *"Is the oil filter of the turbine gearbox clogged?"*, Fault detection, **(SPARQL)**; **Q5.** *"Is the wind turbine component thermo-bypass valve faulty?"*, Fault detection, **(STAR, SPARQL)** (cf. Sec. 4.2); **Q6.** *"How are all the different sensor and component faults occurring within same time period related?"*, Fault diagnosis, **(SPARQL, STAR)** (cf. Sec. 4.3)

5.2 Evaluation Results

Semantic data encoding and loading. The number of triples generated by the RDF-encoder plus the TopSPIN rule engine for fault and component state pre-detection in the semantic sensor data sets, as well as the sizes of the materialization of these data sets in the triple store SwiftOWLIM are as follows:

Period	1 day	3 days	5 days	7 days
Encoding	185861	557381	928901	1300421
Rules	12692	40037	82073	128111
Material.	373044	1118718	1881619	2648509
Total	571.597	1.716.136	2.892.593	4.077.041

For subsequent observation periods within one week, the size of the materialized sensor data set increases with a factor of about 2.9 on average. Figure 6 shows the times needed for (a) semantic encoding (yellow bar), and (b) pre-detection of faults and component states with SPIN rules (green bar), for different sizes of sensor data volumes.

The times needed by the SDRE to load such extended semantic sensor data sets for semantic query answering are shown in the figure 6 as well. Please note that this includes the loading times for all three of its functional modules SwiftOWLIM, Fact++, and STAR. The increase of the loading time in relation to the increasing size of the semantic sensor data set is due to the data set extension by the SPIN rule engine during the pre-detection phase. The total time for encoding, pre-detection and loading the sensor data recorded for one fulll week (4M triples) to about one hour, which was accepted by our customer. The internal precomputations performed by the Fact++ reasoner while loading the sensor data set are concerned with *class assertions, class hierarchy, data property assertions, object property assertions, object property hierarchy and consistency checks.*

Query response times. The response times for the test queries mentioned above were measured (in seconds) after loading of the respective semantic sensor data set was complete.

Period	1 Day	3 Days	5 Days	7 Days
Q1	0.001	0.01	0.021	0.03
Q2	0.017	0.016	0.032	0.016
Q3	0.327	0.937	1.667	2.495
Q4	0	0.875	1.334	1.76
Q5	9.71	58.378	122.428	189.442
Q6	3.17	2834.201	5553.004	8415.513

The queries Q1 and Q2 are evaluated within less than a second and only against the concept base (TBox) of the domain ontology which never changed during the testing. The FCM system of HYDAC has no fully functional redundant sensors which the analysis component correctly checked (Q1). Query Q3 makes extensive use of the UNION keyword to capture alternative structures, which explains the significant increase of response time for increasing data sizes. For queries Q4 and Q5, the large increase is due to the insertion of detected component faults to the triple store by the CONSTRUCT-part of the respective SPARQL queries, though one could factor the time of the materialization out which is triggered by this addition of triples. Adding new explicit triples monotonically extends the infered closure during materialization and hence increases the overall response time. For query Q5, we see a jump in response time starting from the data file for 3 days which included the fault of the thermo-bypass valve. The contribution of the STAR relational query response time to the overall query response time is minimal; STAR relational queries are evaluated over a fixed graph which is initialized at loading time and very small in comparison to the whole ontology. Even though the underlying problem is characterized as the Steiner tree problem, which is known to be NP-hard, additional tests showed that the response time for STAR queries was negligible in our experiments (STAR provides a polynomial-time algorithm to find approximated results). However, the SPARQL query part (not the STAR query part) of Q5 still can be optimized because its response time increases even though the TBV component faults occurred in the 3-day observation period (at day 3) and remains the same for five and seven days. The most complex test query Q6 is concerned with finding the semantic relations between all sensor and component faults which occurred at the same time. That yielded a x1000 query processing time increase for the test data from day 3 on, since on this day the first faults were documented in the data set. For example, there was one (TBV) component fault lasting from day 3 to day 7 which was detected by SwiftOWLIM by processing the SPARQL query of user query Q5 for every observation per minute in the data set. As a result 55616 triples were added in total to the fact base with the SPARQL CONSTRUCT-part of Q5 which was executed before Q6. Similarly, the oil filter blockage (component) fault just on day 5 caused the addition of 549 triples for the respective component fault type.

We conclude that our proposed system can handle 7 days of observed multi-sensor data of wind turbines while maintaining a reasonable runtime for semantic analysis. This is also considered as feasible time span and response time by FCM experts and thus meets their requirements regarding such a system. In general, the given set of analysis queries might also be answered with a combination of other traditional database and information system technologies. However, especially the semantic inference of implicit knowledge in the RDF-encoded sensor data and the determination of semantic relations between a given set of sensors and components based on the recorded sensor data are crucial for the ICM-Wind application. For this purpose, semantic technologies are a first-class candidate to adopt. Besides, none of the currently available FCM systems in the domain which employ other technologies (cf. Sect. 6) is supporting the engineer in terms of answering the customer-defined set of FCM-related fault detection and diagnosis queries as shown for simple examples above.

6. RELATED WORK

In general, our work on the ICM-Wind system for intelligent FCM of wind turbines is most related to work on intelligent CM and semantic sensor networks. Alternatively, some recent approaches to low-cost condition monitoring of wind turbine gears are only monitoring the output power and rotational speed without even using any FCM sensors [16] to reduce additional maintenance costs caused by, for example, defective sensors.

However, to the best of our knowledge, there is currently no approach for FCM of wind turbines available which makes use of semantic technologies for this purpose. As mentioned above, current FCM systems in the domain focus on fault detection based on multi-variate statistical analysis of sensor data [17]. There are a few approaches to intelligent CM which employ neural networks [1, 11], intelligent agents for distributed data interpretation [12] and semantic reasoning [7] for fault detection and diagnosis, but none are available for intelligent FCM of wind turbines yet. One notable and prominent example is the commercial TIGER system for gas-turbine condition monitoring [14] which uses fuzzy causal graphs to describe models of normal behavior of turbine components and model-based reasoning to detect and diagnose abnormal behavior (faults). In particular, it diagnoses a faulty component through tracing backward in its causal graph to measured parameters and returns the subcomponents on the resulting paths as potential influences which are assumed to be at the origin of the component's misbehavior. This is to some extent similar to the type of fault diagnosis performed by the ICM-Wind system with the semantic object-relational reasoner STAR. Recently, [7] proposed to use semantic technologies in support of condition monitoring and maintenance of machinery in general. In particular, an upper level ontology for CM in OWL and an abstract system architecture for semantic query answering with SPARQL and rule reasoning with Jena is described. Though the architecture is, in principle, similar to the semantic analysis component of our ICM-Wind system, the proposal has neither been implemented nor used for any CM or FCM application yet. A large body of work on semantic sensor networks (SSN) is related with respect to the modeling of sensor data with ontologies and the querying of semantically encoded data for a given application. For example, recently [2] proposes an approach to ontology-based sensor data and metadata querying in large-scale sensor network using the GSN middleware and SPARQL$_{stream}$. The semantic sensor data analysis performed by the ICM-Wind system differs from this and, to the best of our knowledge, other work in the SSN area mainly in the following aspects: The semantic analysis (a) relies on a new, specific domain ontology for FCM of wind turbines, (b) combines different kinds of semantic reasoning where appropriate to answer the FCM-related queries for fault detection and diagnosis given by our customer, and (c) is combined with the statistical LDA analysis when required.

7. CONCLUSION

This paper presented the first system, called ICM-Wind, for semantics-empowered fluid condition monitoring (FCM) in wind turbines, with particular focus on its application of semantic technologies. We showed how the semantic analysis component exploits different means of semantic reasoning and query answering either individually or in combination in order to answer a given set of types of FCM-related analysis queries as required and with reasonable response times. The prototyped system was successfully tested by the HYDAC Filter Systems GmbH based on selected FCM multi-sensor and operational data for two wind turbines. Ongoing work is concerned with the integration of a component for semantic sensor data stream reasoning and an extension of the hybrid analysis component for FCM-related query answering.

Acknowledgements. The ICM-Wind system was developed and evaluated in practice in collaboration with Ajatullah Guenel and Torsten Bley from ZeMA GmbH, and Joerg Kleber, Alexander Wohlers, and Horst Mannebach from HYDAC Filter Systems GmbH, Sulzbach, Germany, with support by the ABO Wind AG, operator of the Windpark Marpingen, Germany.

8. REFERENCES

[1] Andersson, C.; Witfelt, C. (2000): Advisor: A Prolog implementation of an automated nerual network for diagnosis of rotating machinery. http://www.fcss.ukma.kiev.ua/courses/IN.B.09/VIP52PE/HTML/vip/articles/carstenanderson/

[2] Calbimonte, JP.; Jeung, H.; Corcho, O.; Aberer, K. (2011): Semantic sensor data search in a large-scale federated sensor network. Proc. 4th Intl. Workshop on Semantic Sensor Networks.

[3] Compton, M.; et al. (2012): The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics*, 17.

[4] Guenel, A.; Meshram, A.; Bley, T.; Schuetze, A.; Klusch, M. (2013): Statistical and Semantic Multisensor Data Evaluation for Fluid Condition Monitoring in Wind Turbines. Proc. 16th Intl. Conf. on Sensors and Measurement Technology, Germany.

[5] Hameed, Z.; Hong, Y.S.; Cho, Y.M.; Ahn, s.H.; Song, C.K. (2009): Condition monitoring and fault detection of wind turbines and related algorithms: A review. *Renewable and Sustainable Energy Reviews*, 13 (1).

[6] ISO TC 108/SC 5 (2005): Condition monitoring and diagnostics of machines - Data processing, communication, and presentation. Part 2: Data processing. Standard ISO 13374-2:2005.

[7] Jin, G.; Xiang, Z; Lv, F. (2009): Semantic integrated condition monitoring and maintenance of complex system. Proc. 16th Intl. Conf. on Industrial Engineering and Engineering Management.

[8] Kasneci, G., et al. (2009): STAR: Steiner Tree Approximation in Relationship-Graphs. Proc. 25th IEEE Intl. Conf. on Data Engineering (ICDE).

[9] Lu, B.; Li, Y.; Wu, X.; Yang, Z. (2009): A review of recent advances in wind turbine condition monitoring and fault diagnosis. Proc. IEEE Conf. on Power Electronics and Machines in Wind Applications.

[10] Mechefske, C. K. (2005): Machine condition monitoring and fault diagnostics. In: Vibration and Shock Handbook. De Silva, W. (ed.), Ch.25, CRC Press.

[11] Rafieea, J.; Arvania, F.; Harifib, A.; Sadeghic, M.H. (2007): Intelligent condition monitoring of a gearbox using artificial neural network. *Mechanical Systems and Signal Processing*, 21(4), Elsevier.

[12] Rudd, S.E.; Catterson, V.M.; McArthur, S.D.J. (2007): Agent-based technology for data management, diagnostics and learning within condition monitoring applications. Proc. 4th Intl. Conf. on Condition Monitoring.

[13] ter Horst, H.J. (2005): Combining RDF and Part of OWL with Rules: Semantics, Decidability, Complexity. Proc. Int. Semantic Web Conf., LNCS 3729, Springer.

[14] Trave-Massuyes, L.; Milne, R. (1997): Gas-turbine condition monitoring using qualitative model-based diagnosis. *IEEE Expert*, 12(3).

[15] Walford, C.; Roberts, D. (2006): Condition monitoring of wind turbines - technology overview, seeded-fault testing, and cost-benefit analysis. Electrical Power Research Institute (EPRI), Tech. Rep. 1010419, Palo Alto (CA), USA.

[16] Yang, W.; Tavner, P.J.; Crabtree, C.J.; Wilkinson, M. (2010): Cost-effective condition monitoring for wind turbines. *IEEE Trans. Industrial Electronics*, 57(1).

[17] Zhijing, Y. et al. (2011): Intelligent condition monitoring via sparse representation and principal component analysis for industrial gas turbines. Proc. Intl. Conf. on Mech. Eng. and Technology, London, UK.

[18] The Open Geospatial Consortium (OGC) Sensor Web Enablement (SWE): http://www.opengeospatial.org/projects/groups/sensorwebdwg

[19] On-shore wind farm in Marpingen (Saarland, Germany) operated by the ABO Wind AG: http://www.abowind.com/com/operational-management/germany-marpingen-wind-farm.html

[20] Fluid condition monitoring by HYDAC: CM-Expert http://www.hydac.com/de-en/service/fluid-engineering/condition-monitoring/product-program/monitoringcontrol.html