iCM-Hydraulic: Semantics-Empowered Condition Monitoring of Hydraulic Machines

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

Andreas Schuetze, Nikolai Helwig
Center for Mechatronics and Automation Technology (ZeMA)
Saarbruecken, Germany

ABSTRACT

We present the first system, called iCM-Hydraulic, for intelligent condition monitoring (CM) of hydraulic machines that combines statistical, probabilistic and semantic data analysis for fault detection and diagnosis with semantic explanations. The modelling of the domain ontology in OWL2 and the probabilistic domain belief network is based on CM standards and domain expert interviews. Fast fault detection and diagnosis online is performed by the system over a multi-variate sensor data feature stream with statistical fault state classification, semantic symptom detection and diagnosis query answering with C-SPARQL, semantic and probabilistic reasoning in the continously updated belief network. Condition diagnosis queries are also answered offline over the central SwiftOWLIM store with history data. The system prototype was developed for our customer HYDAC Filter Systems GmbH and successfully tested for typical hydraulic aggregates and sensors.

Categories and Subject Descriptors

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

Keywords

RDF stream processing, condition monitoring, statistical classification, belief network

1. INTRODUCTION

Hydraulic machines generate and utilize hydraulic power as hydraulic drive systems of various kinds such as for hydraulic driven heavy equipment, metal presses, automobile brakes, aircraft flight controls, and wind turbine generators. Today, the major strategy of operators of hydraulic driven machinery to ensure its reliable and safe operation is to maintain in particular the embedded hydraulic drive system based on its actual condition, rather than by a fixed scheduled, preventive replacement or mere reactive maintenance. An open-loop hydraulic drive system consists of a hydraulic aggregate which realizes a valve-controlled flow of gas-pressurized oil with a pump from an oil tank through tubes to a hydraulic actuator like a piston that is moved with the generated hydraulic force, and back to the tank. The condition-based maintenance of a hydraulic drive system requires a specific condition monitoring (CM) process which, in general, encompasses continuous data collection, fault recognition and fault diagnosis [12]. There are quite sophisticated condition monitoring systems (CMS) for this purpose from many vendors like Siemens, SKF, Bosch-Rexroth, and HYDAC.

However, current approaches and systems to CM of hydraulic machines [7, 11, 15] perform a quantitative, statistical analysis of measured multi-variate sensor data for various relevant physical and hydraulic fluid parameters. In particular, they 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 indication of the onset of failures together with knowledge-based failure diagnosis support to non-experts goes far beyond current CMS in the field. Only very few approaches to intelligent CM exist which employ means of AI and semantic technologies for semantic data analysis offline [10, 8] but not online, and not for CM of hydraulic machines at all.

To this end, we developed the first intelligent CMS for hydraulic machines, called iCM-Hydraulic, that combines statistical, semantic and probabilistic data analysis for fault detection and diagnosis with customized and understandable explanations to the user. The system was developed for our customer HYDAC Filter Systems GmbH and successfully tested for a given typical hydraulic aggregate and sensors.

The remainder of the paper is structured as follows. After an overview of the system architecture and its components in section 2, we describe the iCM-Hydraulic domain ontology and Bayesian network in section 3. The fault detection and diagnosis query processing by the semantic data analysis component and results of the system performance evaluation are presented in sections 4 and 5. We discuss related work in section 6 before we conclude the paper.

SEMANTiCS '15, September 15-17, 2015, Vienna, Austria

ISBN 978-1-4503-3462-4/15/09...$15.00
DOI: http://dx.doi.org/10.1145/2814864.2814865

2. SYSTEM ARCHITECTURE

Requirements and architecture. The main iCM-Hydraulic system requirements which were given to us by our customer HYDAC are as follows. The system shall monitor the condition of a typical hydraulic aggregate with sensors in order to detect and diagnose four specific types of component failures. It shall provide knowledge-based explanations of detected conditions and faults to experts and non-experts. In particular, the system shall answer a customer-given set of informal, high-level detection and diagnosis queries online over a multi-variate sensor data stream within two minutes, and offline over a central data store for two days of sensor data recordings within one hour.

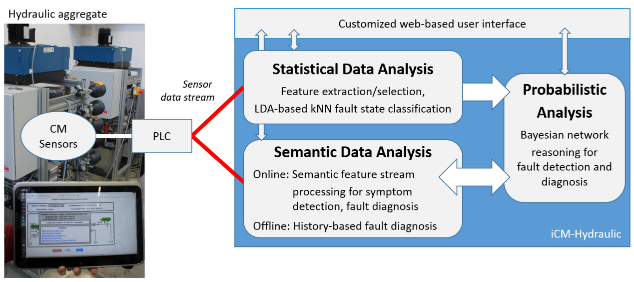


Figure 1: iCM-Hydraulic system components

These user requirements are satisfied by our iCM-Hydraulic system, which component-based architecture (cf. figure 1) is briefly described in the following. In this paper, we focus on the semantics-empowered fault detection and diagnosis with semantic explanation to the user, and refer the interested reader for more details on the statistical data analysis component to [4].

Hydraulic aggregate and sensor data. For testing purposes, our customer provided us with a typical hydraulic aggregate that consists of an oil tank and electric motor pump, which supplies the aggregate with oil as hydraulic fluid, four different gas accumulators with compressed gas for pressurizing the oil, one pressure control valve for controlling the oil flow through the tubes of the aggregate, one proportional pressure relief valve simulating a hydraulic actuator like a piston of a hydraulic drive system, which is driven up and down by the generated hydraulic force, and a cooling unit of the aggregate. In short, the hydraulic aggregate realizes a typical valve-controlled flow of gas-pressurized hydraulic fluid from its tank by its pump in order to move, for example, a piston with hydraulic pressure and back to the tank. This open-loop circuit of the aggregate with its sensors is shown in figure 2; one working cycle of this aggregate lasts one minute.

There are 18 sensors attached to the aggregate, which monitor different physical parameters such as oil pressure and flow rate, vibration, air and oil temperature, metallic and solid particle contamination of the oil, and electric power of the oil pump in the aggregate. These sensors are connected with one Beckhoff PLC (programmable logic controller), which acts as a central data sink and is installed directly on the hydraulic aggregate. The multi-variate sensor data stream is accessible to the iCM-Hydraulic system server from the PLC via a wired EtherCAT connection with a rate of one bucket of raw sensor data of size 694KB per minute. Each data bucket contains 30 measurements from 18 sensors

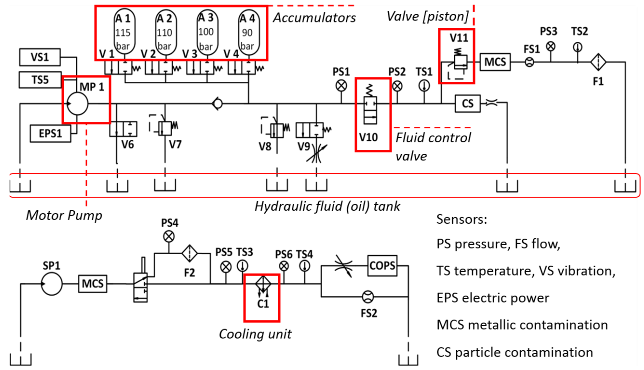


Figure 2: Circuit of hydraulic aggregate with sensors

with in total 50.000 data values per constant working cycle period of one minute of the hydraulic aggregate. The PLC is ordering the time-stamped sensor data stream according to the given stream scheme in JSON. The following fault scenarios for the four main components of the aggregate were manually introduced by an engineer to the aggregate in order to test the performance of the iCM-Hydraulic system: Oil leakage of the pump, gas leakage of gas accumulators, switching faults of the control valve, and decrease of cooling power of the cooling unit.

Statistical fault state classification online. This component determines the fault state of each of the aggregate components with an offline learned kNN (k-Nearest Neighbor) classifier on feature-reduced sensor data. In particular, for each sensor and working cycle, the component extracts the most significant features for fault detection of different components. For this purpose, the feature values of sensor data are computed with respect to signal shape (slope, min, max, position of max) and statistical values (median, variance, skewness, kurtosis). A correlation analysis (Pearson, Spearman) was used to identify those features that are most significant for the studied fault scenarios. The maximal separation of fault grades or fault state classes is done with a supervised linear discriminant analysis (LDA), which performs a linear projection from previously selected features to one discriminant function that allows the quantification of fault severity grades. For fault state classification, the trained LDA-based kNN classifier returns a record with the classified state level for each of the four component fault types but without any explanation of these faults or corresponding component condition.

Semantic analysis online and offline. This component performs a semantic analysis of the sensor data stream from the PLC online and offline and provides the user with specific explanation of detected faults and conditions. For this purpose, it transforms the sensor data stream into a semantic feature stream by means of statistical feature computation and semantic annotation according to a specific domain ontology in OWL2 (cf. Sect. 3). In particular, the size of each stream feature data bucket per minute is 326KB containing about 660 RDF triples. This semantic feature stream is stored by the component in (a) the fact base of its central in-memory SwiftOWLIM triple store, and (b) the fact base of its RDF stream processing engine C-SPARQL. The first is used for history-based fault diagnosis offline and on-demand, while the second is used for the online detection of fault symptoms and external factors, and fault diagnosis. Depending on the type of analysis query, the query processing workflow employs C-SPARQL or SPARQL query answering either individually or in combination with semantic reasoning. The semantic diagnosis results with probabilities are explained to the user in form of text and/or tables according to query-specific templates, which were desgined in collaboration with our customer.

Probabilistic analysis online. This component determines the most likely component faults and conditions for given states of symptoms and faults as evidences provided by the semantic and statistical analysis components. For this purpose, it maintains and employs the probabilistic knowledge about the semantic relations between conditions, faults and symptoms of the hydraulic aggregate components in a domain-specific bayesian network (BN). The conditional probabilities of these relations are initially computed for the given training data set of the statistical fault state classifier, and updated after each symptom and fault detection by the system with or without external (binary) feedback by the user. In addition, the results of probabilistic fault detection and diagnosis are added to the central triple store of the semantic analysis component. The BN component is realized with the open-source BN tool GeNIe.

The iCM-Hydraulic system answers the customer-given set of informal analysis queries by means of query-specific processing workflows, which appropriately make use of different methods of semantic (stream) processing, probabilistic, and statistical data analysis. The knowledge-based explanations are generated according to query-specific explanation templates and displayed in the customized, web-based user interface.

3. SEMANTIC MODEL

The semantic model of the iCM-Hydraulic system consists of two parts: The iCM-Hydraulic domain ontology formally represents knowledge on the condition monitoring domain and its adoption for typical hydraulic aggregates and sensors in OWL2, and the iCM-Hydraulic bayesian network, which represents probabilistic knowledge on the semantic cause-effect relations that are modeled in the domain ontology.

iCM-Hydraulic domain ontology. In particular, the domain knowledge is specified in a formal ontogy in OWL2 under (fully RDFS compatible) OWL-Horst semantics [13] based on the standard vocabularies ISO-2041, ISO-13372, and ISO-17359:2011 for condition monitoring, the results of interviews of domain experts at HYDAC, and a respective extension of the standard W3C SSN (semantic sensor network) ontology. In particular, the concept base of the iCM-Hydraulic domain ontology consists of in total 279 concepts and 184 relations (and 4 XSD data types), which define the semantics of typical hydraulic aggregate components, sensors, measured properties, component faults, symptoms, and conditions, as well as external factors and condition-fault-symptom relations; parts of this domain ontology are shown in figure 3. In support of semantic symptom detection online, the system indexes C-SPARQL queries for instance retrieval of the 20 concepts of fault symptoms, and the state of 6 external factors such as the operational state of the pump or control valve of the aggregate with respective value integrity constraints in the FILTER clauses. The chosen OWL2 fragment was expressive enough for our modeling of the domain, and efficiently implemented with material-

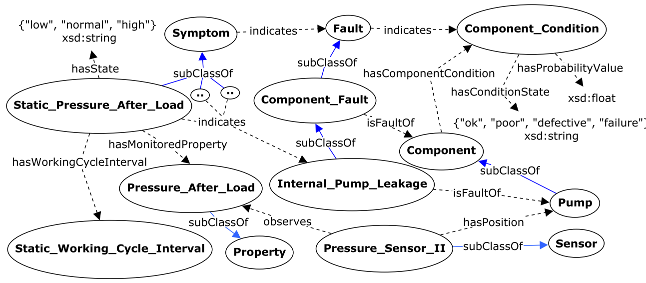
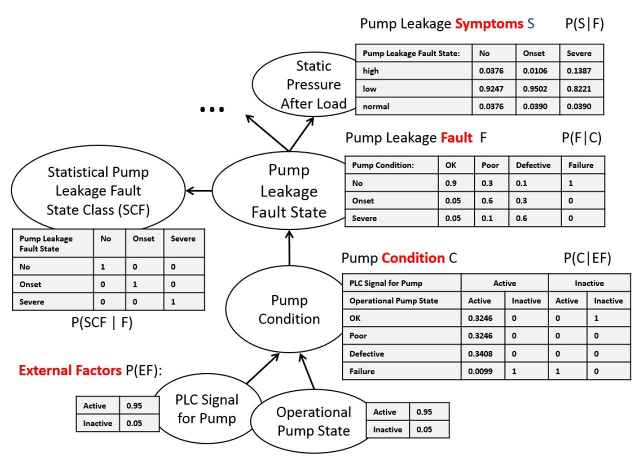


Figure 3: Part of domain ontology with concepts of sensors, faults and symptoms

ization by the in-memory triple store SwiftOWLIM, as well as supported by the RDFS-object-relational reasoner STAR and the reasoner HermiT, which are employed for semantic analysis. The fact base of the central triple store contains the instances of components and sensors, which are particular to the HYDAC hydraulic aggregate along with RDF encoded historical data of sensor measurement and detected conditions, faults, symptoms and probability values as a result of the semantic data analysis online and offline. Semantic fault diagnosis queries in SPARQL are processed offline over this central fact base with history data, while semantic fault symptom detection and diagnosis online is performed over the internal stream fact base of the C-SPARQL stream processing engine (cf. section 4).

iCM-Hydraulic bayesian network. The probabilistic knowledge about causal relations between component conditions, faults, and symptoms are compactly represented in the Bayesian network (BN) of the probabilistic analysis component (in short: BN component) of the system. In particular, the directed acyclic graph maintained by the BN component consists of 36 nodes and 60 relations with conditional probability tables (CPT) attached to them (cf. figure 4).



Pump Leakage Fault State:	No	Onset	Severe
high	0.0376	0.0106	0.1387
low	0.9247	0.9502	0.8221
normal	0.0376	0.0390	0.0390

Pump Condition:	OK	Poor	Defective	Failure
No	0.9	0.3	0.1	1
Onset	0.05	0.6	0.3	0
Severe	0.05	0.1	0.6	0

Pump Leakage Fault State	No	Onset	Severe
No	1	0	0
Onset	0	1	0
Severe	0	0	1

PLC Signal for Pump	Active		Inactive	
Operational Pump State	Active	Inactive	Active	Inactive
OK	0.3246	0	0	1
Poor	0.3246	0	0	0
Defective	0.3408	0	0	0
Failure	0.0099	1	1	0

Active	0.95
Inactive	0.05

Active	0.95
Inactive	0.05

Figure 4: Part of iCM-Hydraulic Bayesian network.

The set of random variable nodes is concerned with 4 component conditions, 4 faults, 20 symptoms, 6 external factors like operational state of the pump, and 4 statistical fault state classifications, while a directed edge denotes the causal influence between two nodes. For example, the various conditions of the pump (ok, poor, defective, failure) influence the conditional probabilities of pump leakage states (no, onset, severe). These, in turn, influence the probabilities of occurrences of various pump leakage symptom states such as the pressure (high, normal, low) at the control valve after load. For example, the BN can be used to determine the most likely state fs of an aggregate component fault F for a given symptom S in state s as evidence by computing $max_{(F,fs)}P(F = fs|S = s)$, while for an initial diagnosis of the detected condition $C = c$ of a user-selected component the system is expected to return the component fault state and the list of symptoms with conditional probabilities $P(C = c|S = s)$ (cf. figure 7). As mentioned above, the initial CPT values of the BN are computed from the given training data set of the statistical fault state classifier, and updated by means of a very fast (1K nodes per minute) exact belief propagation through the BN by the BN tool GeNIe after a detection of symptoms and faults per working cycle.

4. SEMANTIC DATA ANALYSIS

The semantic analysis component of the iCM-Hydraulic system supports the fault detection and, in particular, the online and offline diagnosis and explanation of detected conditions and faults of a typical hydraulic aggregate For this purpose, it makes use of RDF stream processing with C-SPARQL, semantic query answering with SPARQL and reasonig with STAR. For fault detection online the semantic-based symptom detection is combined with statistical fault state classification and probabilistic reasoning, while the statistical analysis is not required for semantic fault diagnosis online and offline.

The analysis query-specific patterns for generating semantic explanations for the user were designed in collaboration with and their instantiation by the system successfully checked by our customer for correctness and usefulness for their personell (cf. Sect. 5).

4.1 Online Fault Detection and Diagnosis

Semantic-based fault symptom detection. The detection of fault symptoms is performed by the parallel processing of 24 symptom and external factor detection queries in C-SPARQL (cf. section 3) over the semantic data feature stream per working cycle of the aggregate. In particular, the symptom and external factor concepts in the domain ontology are reflected in corresponding C-SPARQL queries for their instance detection over the stream fact base.

For example, the operational state of the oil pump is modelled as an external factor and is checked online against the semantic feature stream by evaluating a C-SPARQL query with a respective integrity constraint for relevant sensor data features in the FILTER clause "if `vibration-in-system` $>$ 0.2 and `electric-power-of-pump` $>$ 200 and `accumulator-gas-pressure` $>$ 20) then `pump` = active else `pump` = inactive". The C-SPARQL query for checking the state (low, normal, high) of the symptom `static-pressure-after-load`, which is indicating a pump leakage and its evaluation result for a stream fact base example is shown in figure 5. Since the open-source C-SPARQL engine used by the semantic analysis component does not provide incremental materialization yet, for each symptom concept definition in the domain ontology, the component maintains and uses a special look-up table to identify the required role fillers such as for `isStatisticalFeatureOf` and `hasWorkingCycleInterval` in the example. The created facts for a detected symptom state

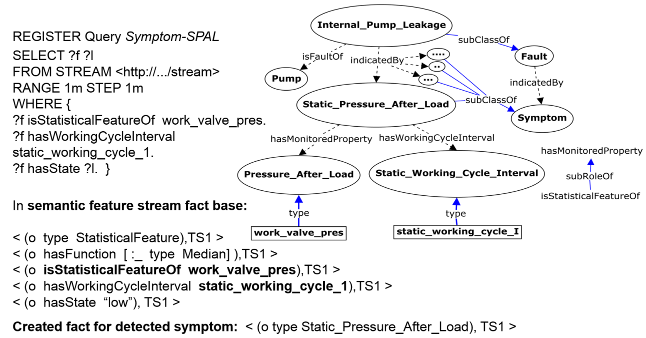


Figure 5: Online detection of symptom

are added to the stream fact base for subsequent fault detection online and the central fact base in SwiftOWLIM for on-demand diagnosis offline.

Hybrid semantic fault detection. The semantic-based detection of symptom states and the statistical fault state classification are performed in parallel over the multi-variate sensor data stream per minute. As mentioned above, the BN component validates the detected fault state classes with the detected symptom state and external factors as given evidences. More concrete, for each fault type, it determines the evidentially most probable fault state $max_{(F,fs)}P(F = fs|S = s)$, which are ideally equal to those reported by the statistical fault state classifier.

For example, the fault state classifier provides the BN component with the (F,fs)-record [pump leakage = onset, valve operation degradation = no, cooling unit operation degradation = no, gas accumulator leakage = no] while the semantic analysis component detected the symptom `static-pressure-after-load` (SPAL) with state "low" and the external factor `Pump-operational-state` (POS) with state "active". If the probabilistic validation yields the value for P(PumpLeakage = onset | SPAL = low, POS = active) as the maximum evidential probability of this fault state with likewisely computed values for the remaining fault types, then the result of the statistical classification is considered as correct, and the semantically detected symptoms and external factors are used to create an explanation to the user. The RDF-encoded result of this online analysis is also added to the central fact base for later offline diagnosis with history data.

In fact, the actual result of this online process combining semantic, statistical and probabilistic data analysis serves as an answer of the condition diagnosis query issued by a user for a selected component such as *"What is the most likely explanation of the detected condition of the pump?"*. The semantic explanation to the user is given as required by our customer in terms of visualized marking (red or green) of the fault state of the pump in the circuit together with its evidential probability and list of related symptoms as shown in figure 6. In particular, the semantic analysis component processes one instance of this query type [Q1] for each of the four components of the hydraulic aggregate in parallel over the semantic feature stream.

Under the assumption that the domain ontology and symptom (and external factors) detection rules are sufficiently fine-grained and correctly modeled for their purpose in practice, this hybrid fault detection process can also compensate

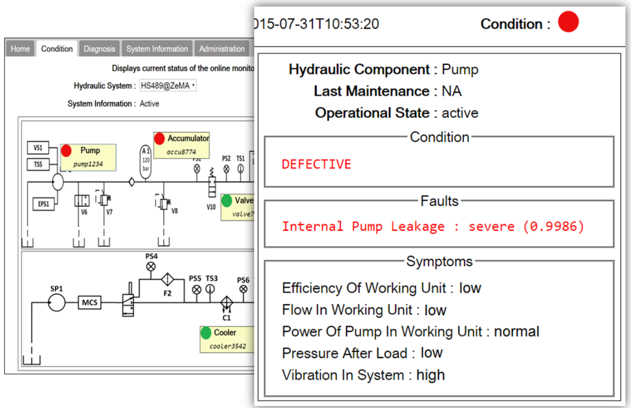


Figure 6: Explanation of pump condition with related symptoms

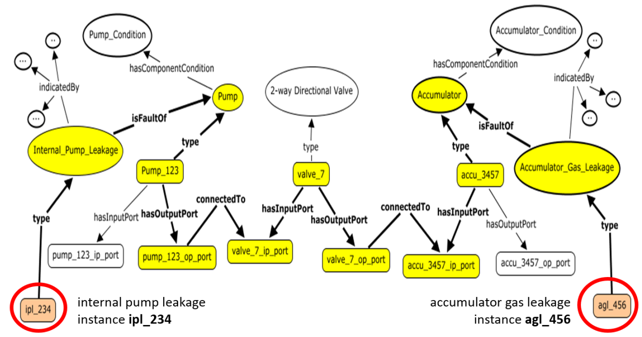


Figure 7: Semantic relation path between faults of pump and gas accumulator

possible statistical misclassifications with presumably correct semantic symptom detection. Suppose that the fault state classifier has been trained offline for fault scenarios of the hydraulic aggregate with oil of certain viscosity, but then oil with a different viscosity is added to the oil tank during regular maintenance. In this example, there is now wrong oil in the system, which is an instance of the respective external factor concept `WOIS` in the domain ontology, but there is no fault of any aggregate component: The statistically misclassified fault state (PumpLeakage = severe) has a lower conditional probability value than the orthogonal but correct fault state (PumpLeakage = no) with the maximum P(PumpLeakage = no | SPAL = low, WOIS = present) for detected symptom and external factor as evidences.

Semantic diagnosis. Other examples of query types for semantic fault diagnosis that were given to us by our customer are *"What is the semantic relation between detected component faults?"* [Q2] and *", which other components are affected by the detected component fault?"* [Q3]. In addition to those four queries of type Q1, the eight queries of type Q2 and Q3 are concurrently processed by the semantic analysis component over the semantic feature stream according to their query-specific processing workflows.

For example, regarding queries of type Q1, the online diagnosis of some detected pump leakage and gas accumulator leakage during one working cycle of the hydraulic aggregate is performed by the semantic analysis component with a combination of C-SPARQL query answering and STAR reasoning. The STAR reasoner [9] creates an internal graph representation of the domain ontology with a non-materialized fact base it then utilizes to process RDF object-relational queries by searching for the shortest paths in this graph between all given objects (query). In general, STAR computes an approximated solution to the corresponding NP-hard Steiner tree problem. The returned property paths between the component fault objects in the ontology are analysed with simple location rules to identify the relative positions of the components in the circuit of the hydraulic aggregate.

Figure 7 shows an example of the shortest property path in the domain ontology between two fault instances of type internal pump leakage and gas accumulator leakage. The result of the path analysis with the following location rules for components X,Y: {X `hasOutputPort` a, a ... b, Y `hasInputPort` b} ⇒ (X before Y) and {X `hasInputPort` a, a ... b, Y `hasOutputPort` b} ⇒ (X after Y), is then used to fill in the query-specific template for the explanation text that is displayed to the user. In this example, the explanation text is *"The pump pump123 with internal pump leakage ipl234 is located before the faulty component accumulator accu3457 with gas leakage agl456 detected at time 12.03.2015 23:00:09. Therefore, the detected internal pump leakage might have caused the accumulator gas leakage."*.

The queries of type Q3 are concerned with checking, which of the actual aggregate components may be affected by a given fault: For each component X the semantic analysis component checks whether there is a shortest semantic relation path in the ontology between X and the given fault instance Y via fault related symptoms S1, S2 with shared property P that is measured by, which sensors Z. The aggregate components X are retrieved with a simple C-SPARQL query and the existence of paths to the fault is checked with STAR queries. These paths from respectively affected components C to Y are matched against the pattern [X..S1, S1 `hasMonitoredProperty` P, S2 `hasMonitoredProperty` P, S2..Y]. The results (P, S1,S2) are used to create and classify concepts of involved types Z of sensors in OWL2 ($QC \equiv Sensor \sqcap \exists observes.(P \sqcap \exists monitorsSymptom(S1 \sqcap S2))$) into the ontology and to retrieve their instances with the reasoner HermiT. Finally, the most likely condition cs for each affected component C with given fault state $Y = fs$ are computed with $max_{C=cs} Pr(C = cs|Y = fs)$.

For example, a detected pump leakage may have affected the actual condition of gas accumulator accu345 and control valve valve987 according to the domain ontology with respective probabilities based on the actual fault symptom related pressure and temperature measurements by sensors of type `PSensorValve` and `TSensorAccu`. This explanation is given in form of the following table as requested by our customer:

Detected fault: Pump leakage onset [0.7]

Affected Component	Condition	Sensor
Accu345	Poor [0.9]	TSensorAccu PSensorValve
Valve987	OK [0.7]	PsensorValve

4.2 Offline Diagnosis

The semantic analysis component also offers the on-demand processing of different types of offline analysis queries over its central triple store SwiftOWLIM with history data. The first three of these types [Q4, Q5, Q6] are reformulations of the three types of queries for online diagnosis for given working cycle periods in the past. In addition, the component processes diagnosis queries of the following types: *"What were the conditions and respective probabilities of given component in the past?"* [Q7] *"What is the frequency of fault type occurrence for a given component and criterias in the past?"* [Q8]. For example, the four (component related) queries of type Q7 are rewritten as instances of the SPARQL query type:

```
SELECT ?t ?cond ?condProb ?grade ?gradeProb
WHERE {
 ?comp type <GIVEN_COMPONENT>.
 ?compFault type Component_Fault.
 ?compFault isFaultOf ?comp.
 ?compFault hasFaultState ?faultState.
 ?faultState hasGrade ?grade.
 ?faultState hasProbabilityValue ?gradeProb.
 ?compFault hasTimestamp ?t.
 ?compCondition type Component_Condition.
 ?compCondition indicatedBy ?compFault.
 ?compCondition hasConditionState ?cond.
 ?compCondition hasProbabilityValue ?condProb.
 ?compCondition hasTimestamp ?t.
} ORDER BY DESC(?t)
```

Another example is the offline analysis with queries of type [Q8]. The result of the respective SPARQL query over the central triple store is the set of all fault states of the selected component for given criterias, which is used by the analysis component to calculate the frequency of their occurrences. Currently supported criterias are high fluid temperature level, and high (4nm, 6nm, 14nm) metallic contamination level.

5. EVALUATION

Implementation and setting. The system is implemented as a client-server Java web application using Google web toolkit. It employs the C-SPARQL engine (R0.9.5.1) for RDF stream processing, GeNIe 2.0 for probabilistic inferencing, MATLAB MCR (compiler runtime) 8.3 for executing statistical data feature functions packaged as Java class methods, SwiftOWLIM (owlimlite5.4) triple store as a central semantic repository, STAR and HermiT as semantic reasoners, and the OWL-API 4.0. The evaluation experiments were performed on a desktop PC with following configuration: Intel(R)Core(TM) i7-2600K CPU@3.40 GHz with 16.0 GB RAM, JDK 1.7 with14 GB Max JVM Heap Space, and Windows 7 Enterprise Service Pack 1 OS.

Test data and queries. The stream of raw data of 50K values of 30 measurements from 18 sensors per minute is transformed by the PLC at the hydraulic aggregate into a multi-variate stream of ordered, time-stamped data buckets according to a given stream scheme in JSON. This data stream is directed by the PLC to the iCM-Hydraulic system via Internet connection. Both statistic and semantic components of the system perform the same feature reduction of the data stream, while the latter component also annotates the feature stream according to the domain ontology. Each feature-reduced stream data bucket per minute consists of 660 triples and is of size 326KB instead of originally 694kb. The stream data rate from the PLC to the iCM-Hydraulic system is fixed and can without feature reduction increase up to 2,000 triples per minute. For semantic offline analysis, the raw sensor data set for periods of one day and two days for 1440, respectively, 2880 working cycles of the hydraulic aggregate with introduced four component fault scenarios, is semantically encoded and materialized in the central fact base with in total 1,067,453, respectively 2,134,906 triples.

For testing the system performance in terms of query response time and accuracy, we have been given the following representative, informal queries by our customer for semantics-empowered analysis online (Q1-Q3) and offline (Q4-Q8) that are processed by the system as indicated above:

- Q1: "What is the most likely explanation of the condition of the pump?". (StatClass, C-SPARQL, BN)
- Q2: "What is the semantic relation between detected component faults of pump and gas accumulator?" (STAR)
- Q3: "Which other components are affected by these detected faults?" (C-SPARQL, STAR, DL, BN)
- Q4: "What is the most likely explanation of each occurrence of the pump's condition in the past?" (SPARQL, BN)
- Q5: "What are the semantic relations between these faults detected in the same working cycle in the past?" (STAR)
- Q6: "Which other components are affected by every detected fault in the past?" (SPARQL, STAR, DL, BN)
- Q7: "What are the pump's condition and its faults, which have occurred in the past." (SPARQL)
- Q8: "What is the frequency of pump's fault occurrence w.r.t. high fluid temperature level in system?" (SPARQL)

Performance measures. The average query response time (AQRT) is measured as the average time taken by the system to answer an instance of a given query type over different data sets. Loading time of sensor data recordings into the central fact base is measured as time needed by different modules of the system to be ready for query answering: That includes time needed (a) for the triple store to materialize the data set, (b) the DL reasoner HermiT 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. The accuracy of statistical fault detection, and average precision (AP) of fault diagnosis query answering by the system is measured.

Statistical analysis performance. The fault state classifier of the statistical analysis component has been trained with a set of about 120 million raw sensor data values for 1,250 working cycles and given 4 fault setpoints of the aggregate; the training with leave-one-out cross-validation took about 5 minutes: feature extraction in 5 minutes, feature selection in 0.8 seconds (0.2 seconds per fault), LDA in 0.4 seconds (0.1 seconds per fault). The accuracy of statistical fault state classification was maximal, which is an improvement over previous results reported in [4], and the whole LDA-based kNN classification of fault states took on average 0.5 seconds (query type Q1) and 0.42 seconds (query type Q4) per working cycle of the aggregate.

Semantics-empowered analysis performance. The average time needed to transform one data bucket in the sensor data stream to a semantic feature data bucket in the semantic feature stream by statistical feature computation and semantic annotation per minute is 0.3 seconds. The time for generating the explanation text or table as results of the query-specific processing workflows displayed to the user is only one second on average. In summary, each of the given test queries for semantic online analysis (Q1-Q3) can be answered by the system within one minute on average (cf. table 1).

	AQRT	StatC	C-SPARQL	STAR	DL	BN
Q1	25.5s	0.5s	23s	-	-	1s
Q2	1.6s	-	-	0.6s	-	-
Q3	41s	-	1s	13s	26s	1s

Table 1: Processing times for online diagnosis

The informal query Q1 is processed over the raw sensor data stream and semantic feature stream to detect faults combining different components of the system: Statistical classification (StatC) is performed in less than one second using MATLAB functions, while the stream processing component of system takes most of the remaining time. There are 24 C-SPARQL queries in total to detect 20 symptoms and 4 external factors: on average each C-SPARQL query execution takes less than a second. The probabilistic analysis (BN) takes only one second to update the BN of 38 nodes. With result display, the system takes about 25 seconds on average to answer the informal query Q1. Answering query Q2 takes even less than two seconds on average due to the fast search for semantic paths between detected component faults.

For Q3 evaluation, C-SPARQL query retrieves the component instances of the aggregate from the static part of the stream fact base in one second followed by the STAR reasoning and pattern matching of the resulting paths within 13 seconds. The respective instantiation of the abstract sensor concept template, the classification of these concepts into the ontology and instance retrieval by HermiT takes 26 seconds on average. The computation of the probable conditions of the affected components given the fault as evidence takes the BN component just one second.

The loading time of the system with respect to sensor data recordings of one day is 15 minutes, and 35 minutes for two days recording. However, the response time for each of the given test queries for semantic offline analysis (Q4-Q8) by the system over the materialized central fact base is well below one minute on average (cf. table 2). The processing

		StatC	SPARQL	STAR	DL	BN
Q4	1-day	0.43s	23s	-	-	1s
	2-days	0.43s	25s	-	-	1s
Q5	1-day	-	0.32s	0.81s	-	-
	2-days	-	0.36s	0.83s	-	-
Q6	1-day	-	0.12s	12s	26s	1s
	2-days	-	0.13s	12s	27s	1s
Q7	1-day	-	0.30s	-	-	-
	2-days	-	0.35s	-	-	-
Q8	1-day	-	0.21s	-	-	-
	2-days	-	0.29s	-	-	-

Table 2: Processing times for offline diagnosis

of the informal query Q4 is the offline execution of Q1 on history data. The maximum time taken by the system for Q4 is during the corresponding SPARQL query evaluation to detect symptoms and external factors, while the BN evaluation remains one second as in the online analysis. Q5 is the offline version of Q2 for which a SPARQL query is used to retrieve all occurrences of component faults at the same time (working cycle time stamp) and each STAR query evaluation takes approximately same time as in the Q2 online analysis. Q6 is the offline version of Q3 for which the respective average response times for STAR and BN remain the same, just the SPARQL query evaluation over the central fact base takes less time than for the C-SPARQL query over the stream. The average query response time of Q4, Q5 and Q6 are similar over different data sizes. Processing of Q7 and Q8 is very fast in retrieving historical information from triple store in less than half a second on different data sizes.

The system achieved a maximum average precision (AP = 1) and recall for the above online fault detection and diagnosis queries over random samples of test data of 1,250 working cycles with simulated different fault grades for the considered four component faults. All results of the processing of the semantic symptom detection and diagnosis queries returned by the system for the correctly classified fault states (see above) were found to be correct and complete by the domain experts from our customer. We assume, that the used implementations of STAR, Hermit, GeNIe, C-SPARQL and SPARQL (SwiftOWLIM) are correct, the C-SPARQL query answering is correct for the fixed data rate and complete data bucket per fixed working cycle (window size).

The system can handle the required 1-/2-days of observed multi-sensor data of a typical hydraulic aggregate with sensors while maintaining a reasonable runtime for online and offline semantic analysis. The reported runtimes and diagnosis results with explanations were approved by HYDAC CM experts and met their requirements regarding such a system. In general, the processing of the set of informal analysis queries given by our customer might have been in part also realized with a combination of other traditional database, data stream management and information system technologies. However, the flexible answering of declarative diagnosis queries in case of certain ontology changes, and, in particular, the offline and online reasoning on semantic relations between a given set of machine components, sensors and fault types is crucial for answering these queries. For this purpose, semantic technologies in combination with probabilistic reasoning are first-class candidates to adopt. Besides, none of the current CMS in the domain (cf. section 6) supports the engineer by answering the set of analysis queries given by our customer and with semantic explanations as shown for simple examples above.

6. RELATED WORK

In general, our work on the iCM-Hydraulic system for intelligent condition monitoring of hydraulic machines is most related to work on intelligent CM in this and other engineering domains. For example, [11] presents a signal-based and model-based approach for online condition monitoring of hydraulic machines. The CMS in [15] performs signal analysis and probabilistic reasoning with BN for offline condition monitoring and root cause assessment in pump process operations of hydraulic machines, and utilizes explicit user feedback for sequential learning to increase its performance. [7] presents a first brake fault diagnosis based on a statistical analysis of vibration signals with decision-tree based feature selection and trained SVM-based fault classifier with high accuracy. However, to the best of our knowledge, none of the current CMS for hydraulic machines performs semantics-empowered condition and fault diagnosis. On the other hand, there are a few approaches to iCM in other domains that make use of semantic technologies for fault detection and diagnosis. For example, our prototyped system ICM-Wind [10] for intelligent fluid condition monitoring of wind turbine gears performs semantic sensor data analysis offline by applying semantic technologies for interpreting the state of turbine parts and answering questions related to their maintenance. Similar to iCM-Hydraulic, the specific domain knowledge is encoded in OWL2 and with SPIN rules; given fault detection and diagnosis queries are answered by use of the semantic reasoners Fact++, STAR, TopSPIN rule engine over a central SwiftOWLIM store. Besides, [8] proposed to use semantic technologies in support of condition monitoring and maintenance of machinery in general. 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 functionality is, in principle, similar to the offline semantic diagnosis performed by our iCM-Hydraulic system, to the best of our knowledge, the proposal has neither been fully implemented, nor used for any CM application in practice yet. Further, the large body of work on semantic sensor networks (SSN), stream reasoning or RDF stream processing (RSP) is related to our work. For example, [1] describes an approach to ontology-based sensor data and metadata querying in large-scale sensor network using the GSN middleware and SPARQL$_{stream}$, while the group of O. Corcho et al. investigates the scalability of semantic stream query answering in the cloud based on Apache STORM and Lambda architecture. However, to the best of our knowledge, none of them combines statistical and semantic feature stream analysis with probabilistic reasoning in particular, and for the purpose of intelligent online condition monitoring of hydraulic machines in general.

7. CONCLUSION

This paper presented the first system, called iCM-Hydraulic, for semantics-empowered condition monitoring (CM) of hydraulic machines, with particular focus on its employment of semantic technologies. The system combines statistical, probabilistic and semantic data analysis for fault detection and diagnosis with semantic explanations to the user. We showed how the semantic analysis component exploits RDF stream processing with C-SPARQL, semantic query answering offline with SPARQL, and semantic reasoning online with STAR and HermiT either individually or in combination in order to answer a given set of CM-related analysis queries as required and with reasonable response times. The system prototype was developed for our customer HYDAC Filter Systems GmbH, successfully tested for a typical hydraulic aggregate and sensors, and demonstrated at the Hannover industry fair 2015. Ongoing work is concerned with its commercial deployment and exploitation by HYDAC.

Acknowledgements. The work reported in this paper has been performed in collaboration with the HYDAC Filter Systems GmbH in Sulzbach, Germany. We are in particular indebted to Joerg Kleber and Dr. Horst Mannebach from HYDAC. This work was also supported by the German ministry for education and research (BMBF) in project INVERSIV under contract number 01IW14004.

8. REFERENCES

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

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

[3] A. Guenel, A. Meshram, T. Bley, A. Schuetze, M. Klusch. Statistical and Semantic Multisensor Data Evaluation for Fluid Condition Monitoring in Wind Turbines. *Proc. 16th Intl. Conference on Sensors and Measurement Technology*, 2013.

[4] N. Helwig, E. Pignanelli, A. Schuetze. Condition Monitoring of a Complex Hydraulic System Using Multivariate Statistics. In *Proc. IEEE Intl. Instrumentation and Measurement Technology Conference*. 2015.

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

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

[7] R. Jegadeeshwaran, V. Sugumaran. Fault diagnosis of automobile hydraulic brake system using statistical features and support vector machines. *Mechanical Systems and Signal Processing*, 52/53. 2015.

[8] G. Jin, Z. Xiang, F. Lv. Semantic integrated condition monitoring and maintenance of complex system. In *Proc. 16th Intl. Conf. on Industrial Engineering and Engineering Management*. 2009.

[9] G. Kasneci, et al.. STAR: Steiner Tree Approximation in Relationship-Graphs. In *Proc. 25th IEEE Intl. Conference on Data Engineering*. 2009.

[10] M. Klusch, A. Meshram, P. Kapahnke, A. Schuetze. ICM-Wind: Semantics-Empowered Fluid Condition Monitoring of Wind Turbines. In *Proc. 29th ACM Intl. Symposium on Applied Computing*. 2014.

[11] D. Lovrec, V. Tic. On-line condition monitoring systems for hydraulic machines. *Mechanical Engineering*, 10(1). 2012.

[12] C.K. Mechefske. Machine condition monitoring and fault diagnostics. *Vibration and Shock Handbook*. Ch. 25, CRC. 2005.

[13] H.J. ter Horst. Combining RDF and Part of OWL with Rules: Semantics, Decidability, Complexity. In *Proc. Intern. Semantic Web Conference*, LNCS 3729, Springer. 2005.

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

[15] G. Weidl, AL. Madsen, S. Israelsson. Object-Oriented Bayesian Networks for Condition Monitoring, Root Cause Analysis and Decision Support on Operation of Complex Continuous Processes: Methodology & Applications. ABB Group Services. 2005