Abstract

In smart oilfields, a large volume of data is being generated related to assets, personnel, environment, and other production and business-related processes on a daily basis. Storing vast amounts of data is only justifiable if it leads to the discovery of actionable insights which can then be translated into improvements in operational efficiency and Health, Environment, and Safety (HES) conditions. Smart oilfield data is of high volume, variety, and velocity and can be located in multiple data silos. This presents an urgent need to develop scalable and extensible techniques that can enable domain experts to access data and perform analytics to yield better decisions and results. The focus of this paper is on the process of Asset Integrity Management and the role of Semantic Web technologies for significantly improving decision-making in this domain. The most significant challenges, thus, are to manage the high volumes of data, create a holistic view of asset integrity data, allow intuitive access to the data, and generate insights through an agile system that can be utilized by domain experts without requiring extensive assistance from IT experts. For this, we present the Smart Oil Field Safety Net (SOSNet) system, a Semantic Web-driven platform, that performs integration of asset integrity data, provides simplified querying mechanism for accessing the integrated data and facilitates analytics on top of it to improve efficiency and robustness of the process of Asset Integrity Management.

Introduction

Various processes in the oil and gas industry generate vast amounts of data (Leber 2012). This data usually comes from extensive instrumentation as well as manually generated documents, e.g., notes, work orders, photographs, and drawings. By consolidating, processing, and analyzing such data, experts can find solutions to existing issues, optimize plant operations, and predict and prepare for possible incidents or failure in future. However, manually finding, integrating, and analyzing data for decision making is labor intensive. Finding and integrating relevant data across multiple data silos is even more challenging as studies have shown that domain experts spend 60-80% of their time in just looking for and collecting relevant data for analysis (Kharlamov et al. 2014; Lesslar et al. 1998). Two main reasons for this bottleneck given in
these studies are (i) distribution of relevant data across multiple files and databases and (ii) the reliance of domain experts on Information Technology (IT) experts for the retrieval of relevant data.

Direct access to information would require domain experts to understand database systems and formal query languages. However, enabling domain experts to have faster access to relevant data can lead to faster decision making and quick resolution of current issues. Valuable insights can be gained to prevent potentially hazardous incidents from occurring in the future. This necessitates a solution that not only acts as an effective data management tool that provides linking of relevant data sources but also enables quick and easy retrieval of integrated data, thus reducing the effort put in by domain experts. We present Smart Oil Field Safety Net (SOSNet) system, a scalable and extensible system, that significantly improves Asset Integrity Management decision-making in smart oilfields. SOSNet leverages Semantic Web technologies to facilitate intuitive access to integrated asset integrity data and empower domain experts to focus on analytics that generates insights and yield better decisions and results.

We demonstrate that this Semantic Web-based framework can serve as the foundation for automated Asset Integrity Management systems in smart oilfields, supporting the acquisition, management, access, and processing of asset integrity data. The contributions of the presented work are as follows:

- **Semantic Model for Asset Integrity Management:** We present a hierarchical semantic model that extends existing ontologies with new concepts. The proposed semantic model is extensible to both within the domain of Asset Integrity Management and other processes in the oil and gas industry. This semantic model is the key to automatic integration of multiple asset integrity data streams and for formulating queries to access the resulting integrated data.

- **Easy and Intuitive Querying Framework:** We present an effective framework that enables domain experts, with no knowledge of formal query languages, to issue user-defined SPARQL queries over integrated asset integrity data without IT assistance. This approach is scalable and domain independent.

- **SOSNet System:** We demonstrate the capabilities of the integration and querying framework through the applications of SOSNet system which provides ease of use and flexibility and facilitates decision making for domain experts in the context of Asset Integrity Management.

**Background**

**Asset Integrity Management Workflow**

Asset Integrity Management (AIM) is a safety-critical process on oil and gas facilities. The objective is to keep track of the structural integrity of various assets (e.g., vessels, pipelines) on a facility. Oil companies have workflows and safeguards in place to prevent Loss of Containment (LoC) incidents where fluid contained within an asset is accidentally released in the environment, creating a potentially hazardous situation for the environment, personnel, and other assets (Saeed et al. 2015). Fig. 1 shows a typical Asset Integrity Management workflow. Through metering and surveys conducted on a facility, data in the form of numeric time series, structured and unstructured text, and multimedia content is collected and stored in various databases and files. This spread of information across multiple sources often requires asset integrity managers and experts to manually identify appropriate sources, query them, and integrate relevant data to understand the state of the assets. This process needs to be repeated for different assets and for multiple facilities at a time. The collected data is then analyzed to identify potential risks and necessary actions are recommended or scheduled. All these different steps of the workflow generate data that is crucial for decision making.
Challenges
Due to the inherent heterogeneity of the data sources involved in the process of Asset Integrity Management, providing on-demand access to an integrated view can be challenging (Saeed et al. 2015). Another challenge is knowledge management, which refers to a systematic way of capturing the results of various engineering models and analyses (Soma et al. 2008). These challenges in creating an effective automated Asset Integrity Management system are summarized below:

- **Integrated view of asset integrity data:** For effective decision making, there needs to be a system that presents a comprehensive and continuous view of the assets and processes. Useful information residing in multiple databases and files makes it difficult to gather relevant information for generating actionable insights.

- **Extensibility to new data sources:** Oil facilities are well-instrumented to monitor various processes, not just the state of assets. Applications based on advanced analytics may lead to new factors being incorporated in complex models for characterizing the behavior of assets. Thus any solution that facilitates integration and mining of asset integrity data needs to be extensible to new data sources.

- **Efficient access to information:** It is critical for asset integrity decision makers to have direct access to relevant pieces of information for making informed decisions. However, domain experts may not have adequate skills to search for relevant data in databases and must rely on IT experts for their data needs, causing delays in the workflow.

To address these challenges, we have developed SOSNet, a data-driven Asset Integrity Management system. The work presented in this paper mainly focuses on modeling, integration, and querying aspects of the SOSNet system.

**Semantic Information Modeling for Asset Integrity Management**

Fig. 2 shows the architecture of the SOSNet system. Semantic Web-based ontologies are used to identify AIM-related entities from available data sources (e.g., equipment, facility, work order, and sensor) and model their attributes and relationships. The semantic model is then used for annotating raw data sources and facilitating their automatic integration.
Data Sources
The data sources, for this project, comprise of multiple CSV files, Microsoft Access database files, images, CAD drawings and PDFs. This data consists of the following recorded information:

- Time-stamped thickness measurements (in inches) collected every couple of years (usually after a gap of 5+ years) for multiple Thickness Measurement Locations (TMLs) on all assets on multiple facilities
- Manually assigned corrosion categories (Severe, Medium, Normal) to assets
- Description of problems observed related to assets and recommended solutions
- Images of assets, focusing on areas of concern
- Work orders providing description of requested work and remedial actions performed
- Specification documents including equipment drawings and Piping and Instrumentation Diagrams (P&IDs)

Semantic Data Modeling
The purpose of ontologies is to capture the domain knowledge in terms of concepts (classes), relationships (object properties) and attributes (data properties) so data can be annotated for the machines to understand. For SOSNet, the Smart Oilfield Ontology (SOFOS) (Saeed et al. 2015) has been extended to create the SOSNet Ontology, a Semantic Web-based model for AIM. SOSNet Ontology is defined using Resource Description Framework (RDF) and Web Ontology Language (OWL), the Semantic Web standards for knowledge representation. Similar to SOFOS, the focus in designing this ontology is data-centric. The main objective is not development of the hierarchy of all possible assets (vessels, tanks, motors, compressors, valves) that exist in a typical oil and gas facility. This has already been well-defined by other ontologies (ISO; PPDM). The primary focus with SOSNet Ontology is to model data streams. However, classes that model related physical assets are also needed to identify and integrate the data streams associated with them. For example, different sensors installed on a vessel can be measuring multiple parameters (e.g., pressure, level, temperature). Another set of records associated with the same vessel can be manually generated data such as notes, work orders, multimedia content. Instances of physical assets serve as focal points for grouping such relevant data streams. This way all data recorded for a particular asset becomes integrated to present a complete view of its status.
A subset of concepts and attributes of SOSNet Ontology is shown in Fig. 3. The class Facility is extended from class System of SOFOS which itself is an extension of class ClassOfOrganization of ISO-15962 (ISO) to define a collection of assets. A facility instance can be a rig, platform, or a refinery.

Figure 3—Relationships between select concepts from SOSNet ontology

Classes Equipment and TML represent the physical assets which are the focus of the asset integrity monitoring and prevention procedures. An equipment in the ontology refers to an entity which has been given a unique ID within a facility. Class TML models a specific location on the Equipment to which a data stream can be associated with. This class is specific to the use case of Asset Integrity Management and models TMLs on equipment. Each TML has a time series of thickness values associated with it. A class Survey is used to group together data collected related to an asset on a particular inspection date. A single instance of Survey can present a complete picture of the state of the associated asset based on all the data collected related to it at a specific point in time and multiple such instances constitute a chronological record of the state of the asset. In context of TML, instances of Survey provide historical data on thickness values. For Equipment, instances of Survey provide other relevant information such as photos, assessments, and unstructured text (problems, recommendations). Images associated with the Equipment are modeled through the class Photo and its attributes. All other survey related information is modeled through attributes of the class Finding. One last crucial piece of information comes from work orders which are modeled as instances of class WorkOrder. Work orders keep track of all the maintenance and repair work performed on an asset. In addition to thickness measurements, ambient conditions (temperature, humidity) can also be associated with all assets, depending on the spatial granularity of the available data. This can be done for an equipment, a facility or an entire production area. Such hierarchy can be easily created by extending the class System from SOFOS, as we did for Facility.

Data Integration

Once the semantic model of the data streams has been created, the next step is to generate instances based on the SOSNet Ontology, i.e., import the structured (tabular) and unstructured data in an RDF repository. An RDF graph is represented by a knowledge base of triples. A triple consists of three parts: \(<subject, predicate, object>\). In RDF graph, Uniform Resource Identifiers (URIs) are used as location-independent addresses of entities, both classes (nodes) and properties (edges). Any entity with a URI can be subject, predicate or object. An RDF graph is a set of all such RDF triples (Ning et al. 2009). The nodes of the graph represent entities which are instances of different classes defined in the ontology, whereas edges
represent relationships between entities within or across data sources. Data integration is facilitated by making instances uniquely addressable through URIs. URIs are defined to represent the organizational hierarchy of assets on a facility, e.g., http://www.usc.edu/cisoft/sosnet/TK21A/T1000/PT1000A represents a URI for a pressure sensing device PT1000A mounted on equipment T1000 located on facility TK21A. The benefit of such a scheme is that it allows differentiation between assets, with identical IDs, located on different facilities.

The RDF Generator in Fig. 2 takes raw data files and imports their content into an RDF graph. The files are processed row by row. All values in each row are linked together based on the relationships defined by the ontology. The column names either correspond to a class or a data property. In the former case, the appropriate information from the row is used to create the hierarchical URIs for the entities which are instances of different ontological concepts. In the latter case, a data property is created in the RDF graph with the value taken from the corresponding cell in the row which can be a numeric, string or date value. The example in Fig. 4 shows the RDF representation of a subset of anonymized data associated with equipment T1000. URIs are not shown for readability. It is important to note that URIs (e.g. http://www.usc.edu/cisoft/sosnet/TK21A/T1000/PT1000A) are hidden from the domain experts who work with IDs as per design documents (e.g. T1000, TK21A, PT1000A) instead.

Figure 4—Visualization of asset integrity data in RDF related to a single equipment, T-1000

After integration, the data initially spread across multiple files, becomes available as hierarchically organized linked data. The integrated data is maintained in an RDF triple store and can be accessed via a SPARQL endpoint. As a result, asset integrity experts can issue complex and meaningful queries such as getting the list of all work orders for assets that have been labeled severely corroded. The SPARQL syntax for this query is shown in Fig. 5. This query acquires integrated data coming from work orders and inspection findings data sources, which were previously independent data sources. By issuing SPARQL queries, domain experts can get access to relevant data according to their needs. However, the limitation is that domain experts are not proficient in Semantic Web technologies. To address this problem, we have developed a technique aimed at allowing such non-expert users to directly interact with semantic data without the help of IT experts. This approach is described in the next section.
Accessing Semantic Data

As discussed in Introduction that a desirable feature of an ideal Asset Integrity Management system is to be intuitive enough so domain experts can use it without excessive assistance needed from IT experts. The assumption is that domain experts do not necessarily possess the understanding of query languages, RDF, Semantic Web, and database systems. Moreover, people at different levels (e.g., technician, supervisor, manager) may have different information requirements from the system. Programming the system with pre-defined queries for all possible scenarios and personnel is impractical if not impossible since data needs can vary from person to person and application to application. Maintaining and updating such a library of queries will again require IT assistance. There is a need for a simplified way of querying semantic data where the domain experts issue queries based on their data needs and the system automatically formulates formal queries.

Automatic SPARQL Query Formulation (ASQFor)

Several systems have been proposed in the literature to allow non-expert users to formulate SPARQL queries indirectly using abstractions. A brief survey of such approaches is provided in the section on Related Work. Here, we present Automatic SPARQL Query Formulation (ASQFor) algorithm which allows domain experts to issue SPARQL queries indirectly. ASQFor relies on an ontology (SOSNet Ontology in the case of AIM) and a keyword-based interface to formulate SPARQL queries which syntactically resemble manually written queries by IT experts. The complete pseudocode of the algorithm with a detailed description of each step, complexity analysis and evaluation on 1990 US Census data is provided in (Saeed et al. 2018a). The ASQFor algorithm does not use any pre-defined information and extracts relevant information dynamically from the ontology. It can work with any semantic database that has well-defined schema ontology and is not tailor-made for any specific process within oil and gas domain. This makes the approach domain independent, insusceptible to schema changes, and portable to other use cases without requiring any modifications to the algorithm. Fig. 6 shows an example set of keywords provided to ASQFor and the translation that occurs through the different stages of ASQFor. In this example, certain attributes with corresponding filtering values have been provided. The invocation of ASQFor is done through a single query function that takes key-value pairs. Keys correspond to ontological terms. Values (e.g., "OPEN" and "SEVERE" in the example) are used for filtering. ASQFor automatically traverses the schema ontology and generates and issues the corresponding SPARQL query and returns the results. A single flexible function

```sql
PREFIX sosnet: <http://www.usc.edu/clsoft
/sosnet/sofos#>
SELECT DISTINCT ?facilityname ?equipmentname
?wodescription ?dateissued WHERE{
?facility sosnet:hasEquipment ?equipment.
?equipment sosnet:hasSurvey ?survey.
?survey sosnet:SurveyDate ?surveydate.
?workorder sosnet:DateIssued ?dateissued.
FILTER (?severity = "SEVERE")
FILTER (?status = "OPEN")
}
```

Figure 5—An example of SPARQL query to get work orders for severely corroded assets
to query the repository, solely based on the information in the database and not on how it is organized and stored, can make developing applications easier for programmers.

![Figure 6—From simple keywords to formal ASQFor-generated formal SPARQL query](image)

**Similarity-based Queries**

Based on number of records in a semantic repository, simple user queries can return dozens even hundreds of results, which can lead to information overload. For domain experts, a list of top-k most important assets may be more useful. One measure of importance can be based on the critical status of the assets, based on the assumption that most critical assets require more urgent preventive measurements. This requires importance (or criticality) to be measured in a quantifiable way, so that assets can be ranked based on that. To accomplish this, we leverage the idea behind content-based recommendation systems (Nguyen et al. 2015). For example, given that a user has watched a movie, the system recommends other movies that e.g. may belong to the same genre or have similar plots. The key idea is that similarity is computed with respect to given entity or a set of entities i.e., already watched movie(s). Similarly, we use this notion of pairwise similarity between assets based on their features. The input to the system is a user-provided asset as the search key and then the system generates a list of most similar assets. This allows the user to define the criteria of importance or criticality indirectly. For example, by providing an asset that has "severe corrosion" and is of type "tank", the assets more likely to appear as top results will be other tanks with severe corrosion. We use two approaches for computing pairwise similarity, presented here.

**Similarity based on Attribute-to-Attribute Comparison.** This method of computing pairwise similarity of assets is based on computing similarity between all assets using all attributes, one at a time. The final computed similarity score between two assets is a weighted sum of similarity scores computed based on individual attributes. Fig. 7 shows RDF graph-based representation of two assets. We assert that the similarity of these two assets can be computed by comparing their corresponding direct and indirect attributes. Direct attributes are linked to the asset by a single edge (or predicate) which, in Fig. 7, are fluid, type, and, status. Indirect attributes are those which are direct attributes of other entities (e.g., Surveys, Work Orders, Photographs) linked to the assets. Problem and recommendation from the survey data and description and solution from the work orders represent the indirect attributes of the shown assets. To compute pair-wise similarity, one by one each direct or indirect attribute is selected and its sets of values,
corresponding to both target assets, are compared against each other. This can be visualized as a bipartite graph between corresponding values of the selected attribute.

![Bipartite graph between values of attributes between two assets](image)

**Figure 7**—Attribute-to-attribute based computation of similarity between two assets

Fig. 8 shows the example of computing pairwise similarity based on work orders. The weights of the edges represent the similarity computed between different values of the selected attribute, based on string or document matching techniques. The final similarity score based on single attribute between entity1 and entity2 is average of all the edge weights in the bipartite graph. The final similarity score between two entities based on all attributes is the weighted sum of all single attribute similarity scores. This methodology can be viewed as the top-down search of relevant attributes followed by bottom-up computation of similarity. For each given asset, the algorithm goes deeper (or farther) in the surrounding RDF graph for collecting attributes and their values for computing similarity. The similarity between values of a single attribute (represented by edges in the bipartite graph) are then computed which are averaged to compute single attribute similarity scores. Moving up (closer to the asset), the algorithm computes the weighted sum of single attribute similarity scores to compute the pairwise similarity of assets based on all attributes.

![Bipartite graph between values of attributes between two assets](image)

**Figure 8**—Bipartite graph between values of attributes between two assets

This approach requires the pre-determination of a set of attributes and their weights for computing similarity. In contrast to this methodology, we next present another approach which relies on random walks for automatic collection of values of attributes and uses them to learn vector representations of assets using neural language models in an unsupervised manner.
**Asset2Vec: Generating Propositional Representations of Assets.** In this approach, instead of extracting attributes and their values for computing similarity, representative neighborhoods of each asset are extracted using biased random walks (Cochez et al. 2017; Saeed et al. 2018b). These extracted subgraphs are then converted into feature vectors using neural language model Skip-gram (Mikolov et al. 2013). Fig. 9 shows the workflow of obtaining vector representations for RDF subgraphs. Pruned and biased random walks (Saeed et al. 2018b) are used to traverse the neighborhood around target entities. The extracted subgraphs are represented as a sequence of tokens, resembling a document, as shown in Fig. 9. The neighborhoods of target entities are traversed, up to depth 4, for subgraph extraction. Using these extracted subgraphs, Skip-gram models are trained using the following parameters: dimensions of generated vectors = 500, window size = 10, negative samples = 25, iterations = 5 for each depth. All models for depth d > 1 are trained using sequences generated for both depths 1 and d. Once the feature vectors are generated for all assets, simple cosine similarity measure is used to find the vectors (representing assets) most similar to the given search key.

![Figure 9—Asset2Vec: Generating Propositional Representations of Assets](image)

**Results.** Table 1 shows the top-5 results for equipment similar to given search keyAAQ-0108 (substitute name for actual instrument ID). The lists of most similar assets are generated using both techniques discussed in this section. It can be seen that there are entries that are common to both lists. However, the orders of both lists are not identical. The attribute-to-attribute comparison is based on a pre-selected set of attributes and weights for computing pairwise similarity scores, whereas Asset2Vec selects attributes and labels from the neighborhoods of assets in the RDF graph in an unsupervised way. Therefore, a difference in the order of terms is expected between the two approaches because of difference in their methodologies for selection of attributes. Regarding interpretability, the first approach fares better, since the criteria for computing similarity can be concretely defined through the selection of attributes and corresponding weights. The unsupervised approach is like a black box since upon examining the results it is not immediately clear why certain assets have been declared most similar. A manual inspection of the data, however, revealed that the search key and results share values of one or more attributes from P&ID, equipment type (piping), or corrosion status (Moderate). The benefit of Asset2Vec is that it does not require pre-selection of relevant attributes and corresponding weight assignment, meaning no domain knowledge is required.
<table>
<thead>
<tr>
<th>Search Key</th>
<th>Attribute-to-Attribute Comparison</th>
<th>Asset2Vec</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAQ-0108</td>
<td>AAQ-0103 AAQ-0105 AAQ-0102 AAQ-0105 AAQ-0103 PCT</td>
<td>AAQ-0103 AAQ-0103 PCT AAQ-0102 AAQ-0102 PCT MTD-0204</td>
</tr>
</tbody>
</table>

Representative Applications of SOSNet System

In this section we discuss representative applications that have been built on top of SOSNet to demonstrate the capabilities enabled by the semantic data integration and querying framework.

Querying of Integrated Asset Integrity Data Made Easy

Through ASQFor-based querying interfaces, the end user (domain expert) is exposed only to the attributes residing in the database. These attributes can be selected from the interface and then passed to ASQFor which in turn creates the SPARQL query automatically and returns relevant data. This interface can be generated programmatically based on concepts and attributes in the given ontology. The selection of checkboxes shown in the Fig. 10 corresponds to the sample query of Fig. 5. The domain-independent aspect of ASQFor allows this simple search application to be able to query multiple semantic repositories with hierarchically organized data and well-defined schema ontology, functioning as a portable database search application without requiring any modifications to the algorithm.

![Figure 10](image)

Support for Predictive Maintenance Analytics

To keep track of assets’ status, asset integrity experts maintain a database of thickness measurements of external coatings of assets. These measurements must be above minimum thickness thresholds for safe
operation. Surveyors take measurements after regular periods or calculated intervals, based on the rate of
decrease in thickness between two previous consecutive measurements.

A simple application can be developed that accesses all thickness measurement time series of tens of
thousands of TMLs and based on their trends can extrapolate the time in future when the thickness reaches
critical levels. This way critical assets can be identified and ranked based on the lengths of the time gaps
for scheduling maintenance related activities. An example plot is shown in Fig. 11. Two different forecast
curves are shown based on long term and short-term corrosion rates for a particular TML on tank T1000.
Depending on the size of the asset, it can have dozens to a hundred TMLs and hence that many independent
time series. Automatic ranking of assets based on the thickness levels can alert asset integrity managers in
time to schedule repair and maintenance activities accordingly. The ASQFor query function can be used to
access integrated data for the plot shown in Fig. 11. An invocation of the query function to get all TML
measurements for TML# 1.00 on T-1000 is:

```
query (<Facility, "TK21">, <Equipment, "T-1000">, <TML, "1.00">, <MinThick, ">>, <MeasurementDate,
"">, <ThicknessValue, ">>
```

![Figure 11—Prediction of behavior of TML over time](image)

By running multiple iterations of the above query function with different parameters or by removing
the filter on the keyword Equipment, data for all TMLs on a facility can be gathered. Since ASQFor
is insusceptible to changes in schema ontology, using the query function instead of hardcoded SPARQL
queries in the application eliminates the need for updating the application every time the schema ontology
changes.

**Driving the Development of Effective Visualizations**

Visualization is helpful for collaboration and decision making by presenting data in a comprehensive manner
(Killen et al. 2016). One such example is shown in Fig. 12 where an overview of the facility is provided for
the user. Various charts showing alarms and events have been created. Active warnings and critical alerts
are also shown, clicking on which can take users to detailed pages.
Figure 12—Facility overview screen (anonymized data)

A detailed page for particular equipment is shown in Fig. 13 where information about inspection schedule, an overview of TML trend, and data from the latest survey about the equipment are displayed. Clicking on marked points on the equipment diagram can display trends like the one shown in Fig. 11.

Figure 13—Equipment overview screen (anonymized data)
Due to the integration process, the chunks of data that was previously residing in various individual files and databases have been organized and stored in an RDF semantic repository and can be easily queried through the query function of ASQFor. By using a single visualization screen template (e.g., for the equipment overview screen in Fig. 13) and the query function with different filtering values, development of similar visualization screens can be simplified.

**Related Work**

There are multiple ways of allowing users to query data indirectly. One approach is to allow users to query databases using Natural Language (NL) interfaces (Damljanovic et al. 2011; Wang et al. 2007). Some natural language based approaches limit input to a specific set of rules based on pre-defined sentence structures (Ferre 2014). However, these approaches can work well for simple queries which correspond to a SPARQL query with few triple patterns. For example, "Who directed Interstellar?" maps to a SPARQL query with a single triple pattern. Issuing SPARQL queries that have more complex triple patterns may require the construction of lengthy NL sentences. Processing and accurately translating such sentences can be very time-consuming and complex (Kaufmann and Bernstein 2007). Some approaches avoid the challenges of natural language processing. Some of these approaches guide the user step by step with suggestions of terms from the ontology, formulating queries interactively (Bernstein et al. 2005; Kaufmann et al. 2006). Keyword-based approaches avoid the NL or constrained-NL input by allowing users to formulate queries based on a minimum set of relevant concepts and attributes required to convey query intention; in this case, queries don't need to be syntactically correct. Keyword-based querying has become popular due to search engines like Google where, for example, instead of "Who is the director of Interstellar?", users can ask "Interstellar director" and get the correct answer. The ASQFor approach is a type of keyword-based search.

Semantic Web technologies have been used to model data in various domains including external corrosion monitoring (Saeed et al. 2015) and management of maintenance records (Ebrahimipour and Yacout 2015). Recent advances in the field of Internet of Things (IoT) have led to vast amounts of data being generated at unprecedented rates (Gubbi et al. 2013). Semantic Web technologies can be used to build a layer of semantics, termed as Semantic Web of Things (SWoT), on top of the IoT infrastructure for effective management of data generated by such smart devices and networks (Barnaghi et al. 2012). The key concepts of Semantic Web such as ontologies and Linked Data (LD) can be leveraged to build tools for automatic discovery, semantic modeling, integration, retrieval, and processing of IoT data (Saeed et al. 2017).

**Conclusion and Future Work**

In this paper, we presented a framework for Asset Integrity Management that leverages Semantic Web technologies for data modeling, integration, and access. The presented extensible semantic model can be used for integrating scattered information into a coherent and comprehensive view of the AIM environment for the domain expert, facilitating robust decision making. With ASQFor, data can be retrieved through a simple interface and then can be analyzed using existing or emerging data analytics tools. We use the semantic model, the integrated information bus, and ASQFor as the basis for developing a novel Asset Integrity Management system, SOSNet. Secondly, we discussed an extension to the ASQFor algorithm to enable analytics (e.g., ranking of assets) through simple queries, so that the user can get a reduced result set, i.e., the most relevant results based on certain criteria (e.g., importance, criticality or user-defined).

In future work, we aim to increase the types of SPARQL queries that can be issued through ASQFor. Lastly, our aim is to build more applications to support Asset Integrity Management benefiting from the integrated information bus such as a recommender system that provides solutions to new problems based on historical data, inconsistency detection in TML trend data and its automatic disambiguation (determination of cause) through work/repair and survey records, and ranking of entities (i.e., assets, alarms, or events).
based on criticality at different levels of scope (e.g. on one facility or on multiple facilities within a business unit).

**Acknowledgement**

This work is supported by Chevron Corp. under the joint project, Center for Interactive Smart Oilfield Technologies (CiSoft), at the University of Southern California.

**References**


