Dr. Akram Bou-Ghannam

Foundational Ontologies for Smarter Industries

This IBM® Redpaper™ describes foundational concepts for implementing “smarter” industry solutions that aim to optimize industrial processes and operations, and improve decision making. Smarter solutions harness the power of information and require information models to provide the much needed context for integration, interoperability, and analytics.

This paper has the following sections:

Introduction to Foundational Ontologies

Smarter solutions harness the power of information and require information models to provide the much needed context for integration, interoperability, and analytics. Smarter solutions share the common pattern of Monitor → Assess → Predict → Optimize, and thus demand reference information models (ontologies) for capturing and using observation and measurement data.

Key definition: Ontologies are knowledge representation mechanisms that are explicit formal specifications of the terms in a domain and relationships among them. Ontologies are best suited for representing the information models that are needed to enable smarter solutions.

For more information about ontologies, see *Ontology*, found at:

`http://tomgruber.org/writing/ontology-definition-2007.htm`

The ontologies that are presented in this paper constitute a reference semantic model for capturing observation and measurement data, including sensing and sensor networks, quantity kinds and units of measure, geo-spatial representations, and asset registry information. The reference semantic model that is described in this paper can be easily reused across industries in many solutions.

The specific real-world use case that is described in this paper relates to the domain of environmental analytics dealing with real-time monitoring of environmental data to facilitate early detection of and response to operational events surrounding offshore oil and gas installations. The goal is to obtain a license to operate (especially in environmentally sensitive areas) by managing responsible exploration, drilling, and operations with no harm to the environment. The concepts that are explained in this paper are not specific to the environmental monitoring domain but apply to various domains across industries. This paper is useful to anyone interested in implementing smarter industry solutions by using the advantages of semantic technologies.

Figure 1 shows the type of ontologies that are addressed in this paper as foundational ontologies for Smarter Industries.

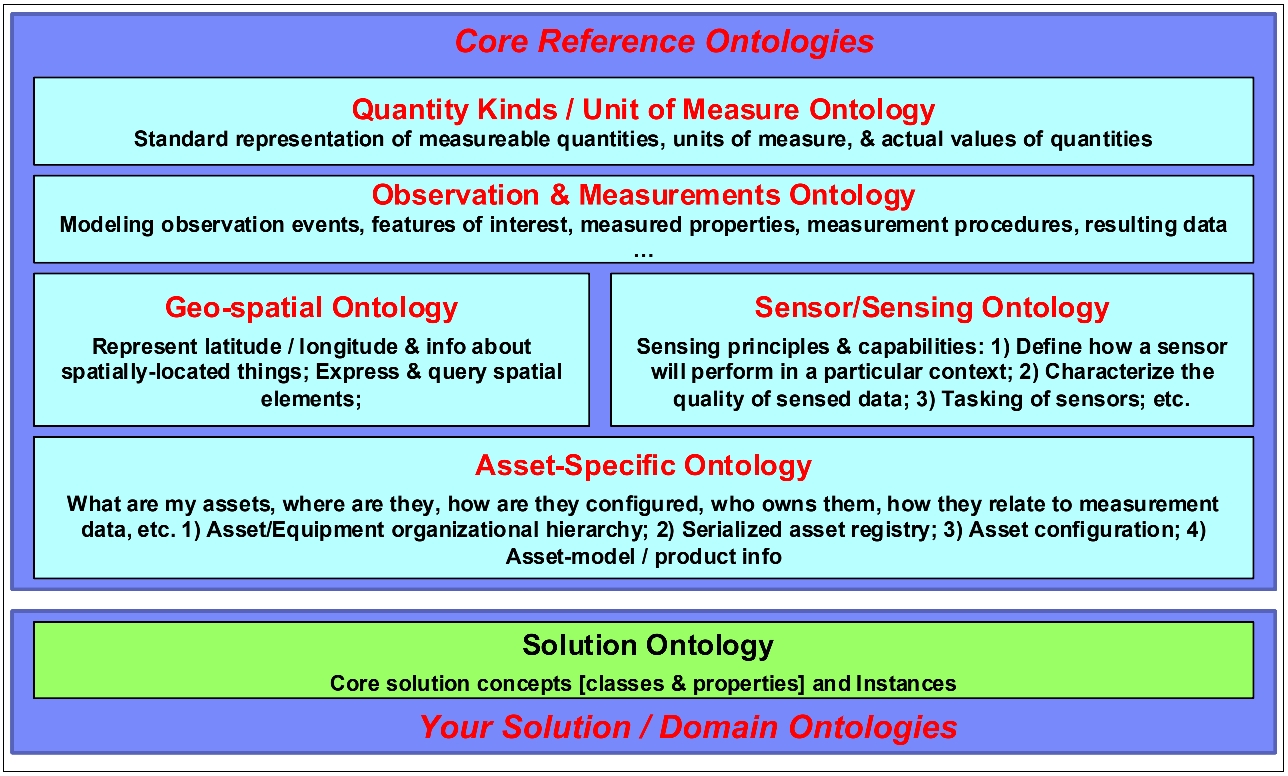

Figure 1 Foundational Ontologies for Smarter Industries

Semantic technology: Better integration and interoperability of diverse information assets

The solution that is described in this paper builds on semantic technology.

Key definition: Semantic technology provides a flexible integration approach that can easily accommodate change with no need to redo the schema. It offers better integration and interoperability of diverse information assets than traditional IT systems. Traditional relational databases are inflexible and fragile when the nature (schema) of the world changes, and thus require constant (and expensive) rearchitecting in the face of new knowledge or new relationships.

For more information, see *Making the Argument for Semantic Technologies*, found at:

`http://www.mkbergman.com/974/making-the-argument-for-semantic-technologies/`

What gives semantic technology these advantages is the Resource Definition Framework (RDF), which is its underlying data model. RDF is a graph data model where new concepts and relationships can be added with no need to change the schema.

Key definition: RDF is the suitable data model for knowledge representation that is adopted by the Semantic Web. It can capture the form, format, serialization, or schema of any existing data. This enables its flexibility to handle a changing world. Any data source or schema can be represented in RDF. The RDF data model provides the canonical means of federating information, which is the foundational benefit of semantic technology.

For more information, see *Making the Argument for Semantic Technologies*, found at:

`http://www.mkbergman.com/974/making-the-argument-for-semantic-technologies/`

Many people think of semantic technology as the technology to enable artificial intelligence (AI) applications, which allows machines to store and retrieve information and infer new facts from existing facts and data, which creates new information. However, outside of the field of AI, there is another complimentary approach for applying semantic technology that focuses on the flexibility of the underlying data model of semantic technology. This characteristic makes semantic technology best suited for data and knowledge management applications, providing significant advantages over traditional technologies. Traditional relational databases are inflexible and fragile when the nature (schema) of the world changes, and require constant (and expensive) rearchitecting in the face of new knowledge or new relationships. Semantic technology highlights a graph data model (RDF) that allows new concepts and relationships to be added without changing the schema. This provides the flexibility to handle a “changing world” and provide better integration and interoperability of diverse information assets. For more information, see *Making the Argument for Semantic Technologies*, found at:

`http://www.mkbergman.com/974/making-the-argument-for-semantic-technologies/`

Today, IBM has an implementation of RDF in IBM DB2® (for more information, see `http://www.ibm.com/developerworks/data/tutorials/dm-1205rdfdb210/`) and it scales well. Linked data standards such as those supported in the Semantic Web are also being applied inside IBM Rational® and Cloud & Smarter Infrastructure products in support of Open Services for Lifecycle Collaboration (OSLC) initiatives (for more information, see `http://open-services.net/)`.

Semantic technology is a key enabler to “contextual computing” and the contextual enterprise. The IBM WATSON Jeopardy! System is an example of contextual computing, and future versions of WATSON will thrive on rich context creation (for more information, see `http://www-03.ibm.com/innovation/us/watson/index.shtml)`.

A use case example: The Environmental Analytics solution

This paper illustrates how the concepts of foundational ontologies are implemented by using a real-world use case from the Environmental Analytics (EA) domain. The solution that is described in this paper can help oil and gas companies monitor and reduce the environmental impact of their operations.

Note: For more information about this solution, see *Smarter Environmental Analytics Solutions: Offshore Oil and Gas Installations Example*, TIPS1131.

EA is an integrated environmental monitoring system that is aimed at helping oil and gas companies minimize the environmental impact of their operations. EA enables real-time monitoring of environmental data, and early detection of and response to operational events surrounding offshore installations. Traditional environmental monitoring techniques are not real-time, but are typically done manually and are labor-intensive.

EA enables the measurement and analysis of physical, biological, and chemical data during actual operations and thus transforms environmental monitoring into an integrated task as part of the day to day operations across the lifecycle of an oil field. This makes it possible for oil and gas companies to predict and more rapidly respond to critical environmental situations. These situations include a stop in drilling, shutting down production at an installation, or ceasing construction activity during environmentally sensitive periods.

EA enables oil companies to implement responsible production operations without inflicting harm to the environment, and thus supports the process of obtaining a license from the authorities to operate in environmentally sensitive areas. The EA solution is built on the semantic framework that is described in this paper, which enables operational business intelligence for various “smart” solutions.

This paper uses the *real-world monitoring of drill cuttings dispersion use case* as an illustrative example to explain how to develop an information model that describes and relates the concepts and entities that are involved in the solution. The information modeling used the existing reference ontologies there were mentioned earlier. The solution model provides the observation and measurement context that is needed for integration, analytics, and optimization. At run time, the various applications (integration, analytics, and optimization applications) use model-driven services to access the needed context. Even though the example that is used here is chosen from the environmental analytics domain within the oil and gas industry, the concepts in this document apply to different domains across industries.

Why ontology-driven smarter solutions

Business drivers call for “smarter” industry solutions to optimize processes and operations and improve decision making, which enables stakeholders to visualize and act upon critical intelligence and events for faster and smarter business decisions. The business requirements aim to enhance production capacity, quality, traceability, operational efficiency, and safety while reducing cost.

A large category of solutions deal with the management and mitigation of operational risk, whether the risk is associated with environmental health and safety, asset health, or quality. The requirement to effectively manage risk demands “smart” solutions to anticipate risk requirements, and run them in advance of critical points to mitigate the risk. This class of solutions typically shares the following common activity pattern:

Monitor → Assess → Predict → Optimize

The solution must monitor important features of interest, and observe and measure selected properties that are associated with the features of interest. Using the captured observation data that results from the monitoring step, along with other relevant data, the solution must then determine the current condition (state) and assess the health of the monitored entities. The next solution step is a prognostics step that determines (predicts) the future health states based on the current health assessment, historical data, and projected usage loads on the monitored entity. The final step provides actionable information regarding the process changes that are required to optimize the life of the monitored entity.

Smarter solutions that are developed to address the business challenges must harness the power of information to deliver their promises. Such solutions must use the correct information models to provide the *context* for integration, interoperability, analytics, and optimization. The information model maintains the context throughout the process, from monitoring to optimization. The information model represents the concepts in the domain and the relationships among the concepts. It is the relationships between concepts (for example associations, constraints, dependencies, and rules) that provide the *context for deeper situational understanding*, which drives better decisions and more effective actions. Context is the key to unlocking the potential of data.

Many industry solutions require an information model for capturing and using observation and measurement data. The observation and measurement context is about maintaining the “What, When, Where, and How” metadata of observations:

- What is being measured?
- When it is being measured?
- Where it is being measured?
- How it is being measured?

This paper describes reusable the cross-industry information models (ontologies) that are used to enable smarter optimized solutions, specifically the ontologies that involve the Observation & Measurement (O&M) model:

- Observation and measurement
- Sensor, sensing, and sensor capabilities
- Quantity kinds and units of measure
- Geo-spatial ontologies
- Asset-specific ontologies (see Figure 1 on page 3)

Ontologies are explicit formal specifications of the terms in a domain and the relationships among them, and are best suited for representing the information models that are needed to enable smarter solutions. Ontologies are explicit declarative models that can be reasoned with by machines, which enable machines to hopefully one day do the thinking for us. Furthermore, ontologies are an essential component of the Semantic Web, where shared reference ontologies are essential to enabling the vision of Linked Open Data.

This paper shares some reference ontologies that were used in real-world engagements. These projects required a reference model for capturing observation and measurement data. As the Semantic Web is all about reuse, this paper describes reference ontologies on observation and measurement, sensing and sensor networks, quantity kinds and units of measure, and geo-spatial representations, which is what you learn about in this document.

The goal is to promote reuse of these reference ontologies and to collaborate to standardize and share the O&M model across industry solutions. The vision includes a common model repository with the harmonized reference ontologies, guidelines, preferred practices, and tools that are shared across industry solutions.

Foundational Ontologies for Smarter Industries overview

Many ontologies exist, but not all of them are suitable for reuse. A critical knowledge engineering task is to select the representative reference ontologies that are appropriate to your domain and scope. This paper presents the reference ontologies that are selected (adopted) for the cross-industry domain of managing observations and measurements. Figure 1 on page 3 shows the core reference ontologies that were used on client engagements, which are the subject of this paper.

The solution domain ontology builds on and extends existing reference ontologies and attempts to reuse knowledge that was developed by domain experts within the solution domain as much as possible. A typical solution requirement is to use the flexibility, extensibility, and openness that is promised by semantic web technologies with open methods for data access. The strategy is to adopt widely used vocabularies and ontologies to define concepts and terms within the solution reference semantic model.

To meet these goals, this paper identifies relevant reference ontologies, including the Semantic Sensor Network (SSN) ontology, the Quantity - Unit - Dimension - Type (QUDT) ontology, and GeoSPARQL and Basic Geo ontologies. The solution domain ontology uses, extends, and harmonizes these reference ontologies, and additionally defines its own concepts, which are not covered by these ontologies.

Quantity kinds and units of measure ontologies

Quantity kinds (for example, temperature, pressure, and velocity) and units of measure (for example, meter, kilogram, and degree Celsius) reference ontologies provide a *standard* representation of measurable quantities, units of measure, and actual values of quantities. These ontologies are needed to provide a unified model of measurable quantities, units for measuring different kinds of quantities, the numerical values of quantities in different units of measure, and the data structures and data types that are used to store and manipulate these objects in software. The ontology includes instance data populating the model with standard quantities, units, and quantity values.

A few reference ontologies for quantity kinds and units of measure exist, each with a different purpose, level of completion, and comprehensiveness. The scope of this paper does not include a comparative study of these ontologies. Rather, this paper focus on the NASA Quantity - Unit - Dimension - Type (QUDT) Ontology (for more information, see `http://www.qudt.org`). This ontology is by far the most comprehensive and complete regarding quantity kinds and units.

Note: For more information about the Quantity - Unit - Dimension - Type (QUDT) ontology, see “Additional information for the quantity kinds, quantity values, and units of measure ontologies” on page 23.

The QUDT ontology defines the base classes properties and restrictions that are used for modeling physical quantities, units of measure, and their dimensions in various measurement systems. The QUDT ontology is a schema ontology and uses the name space and the prefix qudt for all internally defined resources that are described at the following website:

`http://www.linkedmodel.org/catalog/qudt/1.1/index.html`

Observation and measurement ontology

The observation and measurement specific ontologies deal with the concepts of observing properties, which are facets or attributes of features (or features of interest). This ontology models observation events, features of interest, measured properties, measurement procedures, and the resulting data.

The goal of an observation might be to measure or otherwise determine the value of a property. A measurement is a set of operations to determine the value of a quantity, and a “measure” is the value that is described using a numeric amount with a reference scale. So, “measurement” can be reserved for cases where the result is a numeric quantity, and observation is used for the general concept. The observation result is an estimate of the value of a property that is determined through a known observation procedure that is used in making the observation.

All of these concepts and how they are related are the essence of the observation and measurement ontologies. The observation and measurement ontology that is proposed in this paper is specified in the Semantic Sensor Network (SSN) ontology, which has its roots in the Open Geospatial Consortium (OGC) observation and measurement data model. For more information, see the following website:

http://www.opengeospatial.org/

OGC defines Observation and Measurement (O&M) standards (initially, a UML model).

Note: For more information about the O&M ontology, see “Additional information for the Observation and Measurement ontology” on page 22.

SSN defines the O&M model in a formal ontology (in Web Ontology Language (OWL)) by focusing on sensors, the sensing process, and sensing capabilities.

A relevant document from OGC is *Observation and Measurement - Part 1 - Observation schema*, found at:

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CCOQFjAA&url=http%3A%2F%2Fportal.opengeospatial.org%2Ffiles%2F%3Fartifact_id%3D22466&ei=qGy4UfmvPIOG9QTY44DICA&usg=AFQjCNHRMOMN4FRklzFBHYlDy47TU4ZgtQ&sig2=-ysewv-LECutX3UDNtVc6A

This document specifies the core Observations and Measurements model.

Sensors and sensing ontology

The sensor-specific subdomain ontology defines the capabilities of sensors and sensor networks, and covers sensing principles and capabilities, such as the following ones:

- Define how a sensor performs in a particular context
- Characterize the quality of sensed data
- Tasking of sensors

To represent these capabilities, use the Semantic Sensor Network (SSN) ontology (for more information, see http://www.w3.org/2005/Incubator/ssn/ssnx/ssn). The SSN ontology covers the subdomains that are sensor-specific, such as the sensing principles and capabilities, and can be used to define how a sensor performs in a particular context to help characterize the quality of sensed data or to better task sensors in unpredictable environments.

Geo-spatial ontology

Geo-spatial ontologies represent information about spatially located things, and enable a solution to express and query spatial elements. Use the GeoSPARQL and basic Geo ontologies to provide the spatial context, that is, provide a standard way to express and query spatial elements in RDF.

GeoSPARQL (`http://www.opengeospatial.org/standards/geosparql`) is a standard for representation and querying of geospatially linked data for the Semantic Web from the Open Geospatial Consortium (OGC). The definition of a small ontology based on well-understood OGC standards is intended to provide a standardized exchange basis for geospatial RDF data, which can support both quantitative and qualitative spatial reasoning and querying with the SPARQL database query language.

The W3C Basic Geo (WGS84 lat/long) vocabulary (`http://www.w3.org/2003/01/geo/`) is used to represent longitude and latitude mapping / location data in RDF.

Ontology-driven solution architecture

Figure 2 depicts an architectural pattern for implementing ontology-driven solutions. Reference ontologies are the meta-models that provide applications (solutions) with the much needed context for integration, interoperability, and analytics. Applications do not need to be concerned with the various and often proprietary formats of information sources and their specific access API or protocol. Instead, applications use reference ontologies as the meta-model to get access to the data they need through standard uniform interfaces that are compatible with Web-Oriented Architecture (WOA) and the Semantic Web.

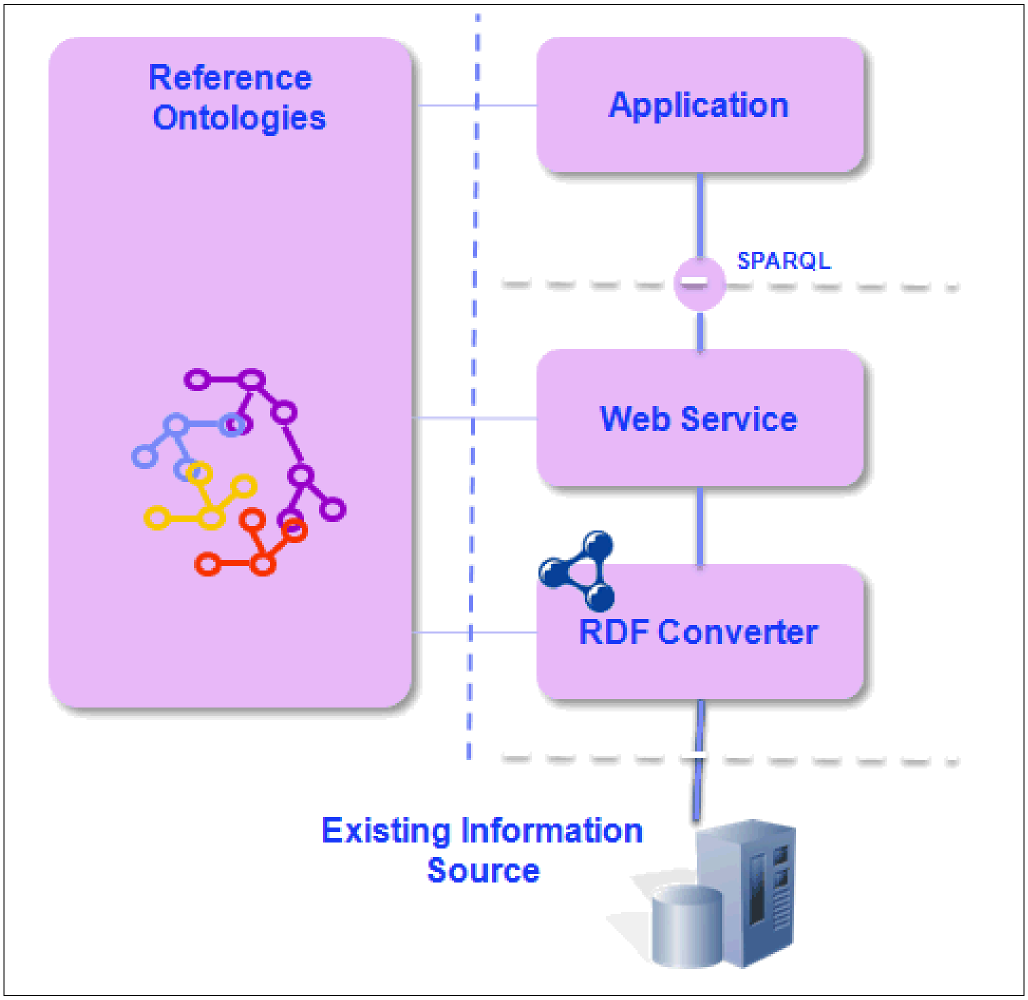

Figure 2 Ontology-driven solution architecture

The architectural pattern is a layered architecture with components in four different layers:

1. Existing information sources layer
2. Conversion / adaptation and access later
3. Ontology layer
4. Application layer

Existing information sources are the existing assets within or outside the enterprise, and typically contain data in diverse formats and can include structured, semi-structured, or unstructured data. With the reference ontologies as the guiding structures, the subset of the content in the information source (which is needed by the solution) is converted into an RDF representation. The converted information is made available to the application layer of the architecture through a set of web services that Web-Oriented Architecture (WOA) principles, and thus enable distributed and loosely coupled access. The web services can expose a SPARQL endpoint for applications to query the information in a standard declarative fashion using terms and concepts that are defined in the reference ontologies. The applications in this architecture are thus ontology-driven and they integrate, search, and manage the information that is made available by the web services layer by using the ontologies as the reference model.

Usage scenarios

The scenarios that are described in this document relate to environmental monitoring for the oil and gas industry, and focuses specifically on the monitoring of the drill cuttings discharges during drilling operations in undersea environments.

Environmental monitoring for oil and gas operations

In general, environmental monitoring describes the processes and activities that must take place to characterize and monitor the quality of the environment. Environmental monitoring is used in the preparation of environmental impact assessments, and in many circumstances in which human activities carry a risk of harmful effects on the natural environment. All monitoring strategies and programs have reasons and justifications that establish the status of an environment or establish trends in environmental parameters. The results of monitoring always are reviewed, analyzed statistically, and published. The design of a monitoring program must consider the final use of the data before monitoring starts.

Environmental monitoring is traditionally a work-intensive and costly phase in impact studies. EA signifies a paradigm shift in environmental monitoring from “expeditionary” offline sampling to continuous environmental monitoring, with sensing devices that provide the data that is needed. The traditional methods are not adequate for monitoring the ecosystems in places where oil and gas activities typically takes place. This method is not viable in the long term, and data must be collected in different ways.

Environmental Analytics (EA) is about monitoring the environment around the oil and gas assets (for example, an oil field) throughout the asset's lifecycle before exploration drilling, through production, and for a specific period after the field is plugged and abandoned. EA provides the capability to measure key parameters across the lifecycle of an oil and gas asset, and to build an integrated system that enhances situational awareness of the environmental impact of oil and gas operations. The system helps optimize operations by turning sensor data and other information into insights that lead to better decisions. The EA system manages all layers of information abstraction from raw measurement data in sensor networks to improved operational decisions and actions. EA helps enable responsible production operations without inflicting harm to the environment.

Environmental Analytics: Monitoring the drill cuttings scenario

Multiple use cases are covered by EA, including the discharge of drill cuttings, and leak detection. This paper focuses on the discharge of drill cuttings use case. Drill cuttings are produced as the rock is broken by the drill bit advancing through the rock or soil. Drill cuttings that are produced by conventional drilling are released either directly from the well (bore hole) to the seabed, or at the seabed, a certain distance away from the well, using a Cuttings Transport System, or at the sea surface.

Here are two of the predominant risks for environmental impact in this case:

- Burial of biological resources (for example, coral) by the drill cuttings sedimentation on the seabed.
- Suspended particles in the water column.

If drilling takes place at locations that have vulnerable biologic resources present, such as coral or sponges, drill cuttings are most often transported onshore for handling. Exemptions from this solution can be made on the condition that real-time environmental monitoring is done during drilling operations to prove that no harm is done to the biologic resources. Different methods can be used to document the impact of discharges:

- Sediment samples using barium as a tracer
- Visual assessment using Remotely Operated Vehicles (ROV)
- Turbidity measurements that indicate particle concentration
- Water current measurements (velocity and direction) at certain depths
- Sediment traps collecting particles that are analyzed for grain size distribution, barium, and other metals
- Image analysis of surrounding areas, such as coral reefs

When used with prudent operations in sensitive areas, environmental monitoring can help reduce the cost that is related to the handling of drill cuttings. The drill cuttings use case makes critical environmental data available both during the planning of the drilling phase and during the drilling operations phase. The goal of this use case is to enable environmental data to become an essential part of drilling operations decisions along with drilling data. To fulfill its goal, it is critical to make the correct type of data available at the correct time with sufficient frequency to enable visualization and analysis of the data.

The EA system uses sensors and real-time communications systems to capture environmental data in real time. The sensors are attached to a subsea platform (called *a Lander*) and include various subsea sensors, such as an ADCP (measures current speed and direction at different heights), a Conductivity-Temperature-Depth (CTD) sensor, camera, echo sounder, fluorometer (a device that is used to measure the parameters of fluorescence, which is the intensity and wavelength distribution of the emission spectrum after excitation by a certain spectrum of light), and a turbidity sensor. In addition, sedimentation traps are placed at key locations to collect sediments. These traps are monitored through the cameras and sedimentation levels are detected through image processing.

Time-based data from the sensors is typically collected and stored in a data historian application. Environmental raw data from sensors and pictures from surveys or sedimentation traps are gathered in to various information sources. All information sources can be federated by using RDF in to an RDF store, such as an IBM DB2 RDF data store, and described by the semantic model. The semantic model is used to categorize each data and describe the relationships between them, considering all known sensor types and individual sensors that are used, and all sensor platforms, including their positions, the geographical areas, and their associated hierarchy.

Here are three main users of the EA portals for the drill cuttings use case:

- The environmental coordinators
- The drilling operation managers
- The Health, Safety, and Environmental (HSE) experts

Environmental Analytics domain ontology

The EA domain ontology includes the EA core ontology (see “EA core ontology” on page 12) and the EA instances ontology for the drill cuttings scenario. The core ontology defines the entities and the associated properties that are needed to describe the EA domain concepts.

Consider the concepts that are involved in the EA domain. EA monitors and evaluates the health of the environment where drilling operations take place. The health of the environment is determined by monitoring and measuring the key properties of the environment. The monitored properties are properties of identifiable objects in the environment, which are the features of interest of the EA observations. For example, the sedimentation level (resulting from drill cuttings) on the coral is a property that affects the health of the coral reef. In this case, the coral reef is the feature of interest for observation, and the sedimentation level is the observed property of the coral reef.

The EA ontology uses reference ontologies (such as SSN) and standards (such as OGC's Observation & Measurement) in defining the EA observation model. The model defines an observed property as a property of the feature of interest. An observation is an action with a result that has a value describing some property. The observation binds a result to a feature of interest, upon which the observation was made. An observation results in an estimate of the value of a property of the feature of interest. The observation feature property pattern is useful for capturing the metadata that is associated with the estimation of feature properties. For more information see “Sensors and sensing ontology” on page 8, and “Observations and measurements” on page 14.

EA core ontology

The EA core ontology defines the EA entities and the associated properties that are needed to define the EA domain concepts. Most of the EA general concepts can be found in the existing reference ontologies, such as SSN, QUDT, and GeoSPARQL. The EA core ontology uses these reference ontologies and defines additional EA domain concepts that are mostly subclasses of the reference ontologies. Some of the classes in the core EA ontology are created to harmonize and link the various reference ontologies.

The EA domain concepts include sensors and sensor platforms, which are entities we are interested in observing (features of interest), such as water quality, water current, and environmental features, such as coral reefs. EA is interested in monitoring and measuring properties of these features of interest, as shown in Table 1.

Table 1 Features of interest

Feature of Interest	Properties
Coral Reef	Accumulated Sedimentation
Water Current	Water Current Speed, Water Current Direction, and Water Current Depth
Water Quality	Water Temperature, Water Turbidity, Water Salinity, Water Fluorescence, and Water Particle Concentration

This following sections show how to model all of these EA concepts in more detail by using the reference ontologies that were introduced earlier in this paper.

Environmental features

EA is concerned with monitoring the health of environmental features such as coral reefs, and identifying whether drilling operations have any effects on their health. EA monitors the properties of environmental features and has need to model environmental features as SSN features of interest. Additionally, the geo-spatial location of the environmental features is important to EA, and there is a need to define environmental features as a geo-spatial feature. Given these requirements, the EA core ontology defines the environmental feature as both a geo-spatial feature (geo:Feature) and as an SSN feature of interest (ssn:FeatureOfInterest), as shown in Figure 3.

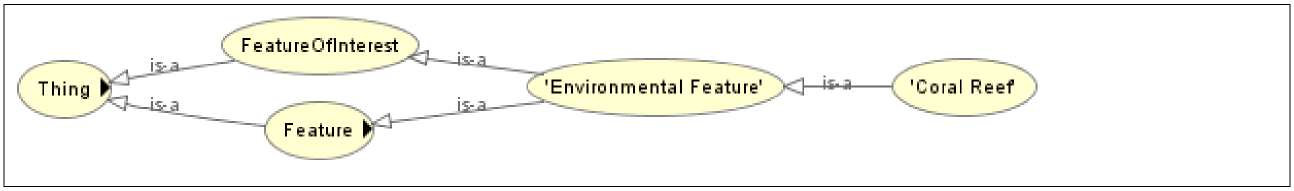

Figure 3 Environmental Features

The geo-spatial location of the environmental features is important to EA. For this reason, this use case defines an *Environmental Feature* class as a geo:Feature (a GeoSPARQL feature):

A geo:Feature is a thing that can have a spatial location, such as a park or a monument. It is related to the geo:Geometry (a representation of a spatial location) through the geo:hasGeometry property (for more information, see the website `http://www.opengeospatial.org/standards/geosparql`).

Modeling an ea-core:EnvironmentalFeature as a geo:Feature provides the geo-spatial location attributes. Example 1 is an RDF representation of environmental feature.

Example 1 RDF representation of environmental feature

```
ea-core:EnvironmentalFeature
        a                     owl:Class ;
        rdfs:label            "Environmental Feature"^^xsd:string ;
        rdfs:seeAlso          <http://purl.obolibrary.org/obo/ENVO_00002297> ;
        rdfs:subClassOf       geo:Feature;
        rdfs:subClassOf       ssn:FeatureOfInterest .
```

A coral reef is an environmental feature that is a feature of interest to EA. The concept of *Feature Of Interest* is defined in the Semantic Sensor Network (SSN) ontology:

ssn:FeatureOfInterest points to the observed feature of interest. A feature of interest can be any observed real-world phenomenon (for example, a geographic entity or entity). For more information, see the ssn:FeatureOfInterest definition, found at:

`http://www.w3.org/2005/Incubator/ssn/ssnx/ssn#term-featureOfInterest`

For more information about the feature of interest concept, see “Sensors and sensing ontology” on page 8.

As EA is interested in monitoring the properties that are associated with the coral reef environmental features, the EA core ontology defines a Coral Reef as an ea-core:EnvironmentalFeature, as shown in Example 2.

Example 2   Coral Reef as an ea-core:EnvironmentalFeature

```
ea-core:CoralReef
        a               owl:Class ;
        rdfs:comment    "Aragonite structures produced by living organisms, found in shallow,
                        marine waters with little nutrients in the water."^^xsd:string ;
        rdfs:label      "Coral reef"^^xsd:string ;
        rdfs:seeAlso    <http://purl.bioontology.org/ontology/ENVO/ENVO%3A00000150> ;
        rdfs:subClassOf ea-core:EnvironmentalFeature .
```

Observations and measurements

EA is concerned with monitoring and measuring various properties of the EA environment. These properties include water current speed and direction, water temperature, water turbidity, and sedimentation levels.

Observations, features of interest, and properties are defined in the SSN ontology, and quantity, quantity kind, and unit are defined in the QUDT ontology developed by NASA. Figure 4 shows the EA core ontology classes that are used to link SSN and QUDT.

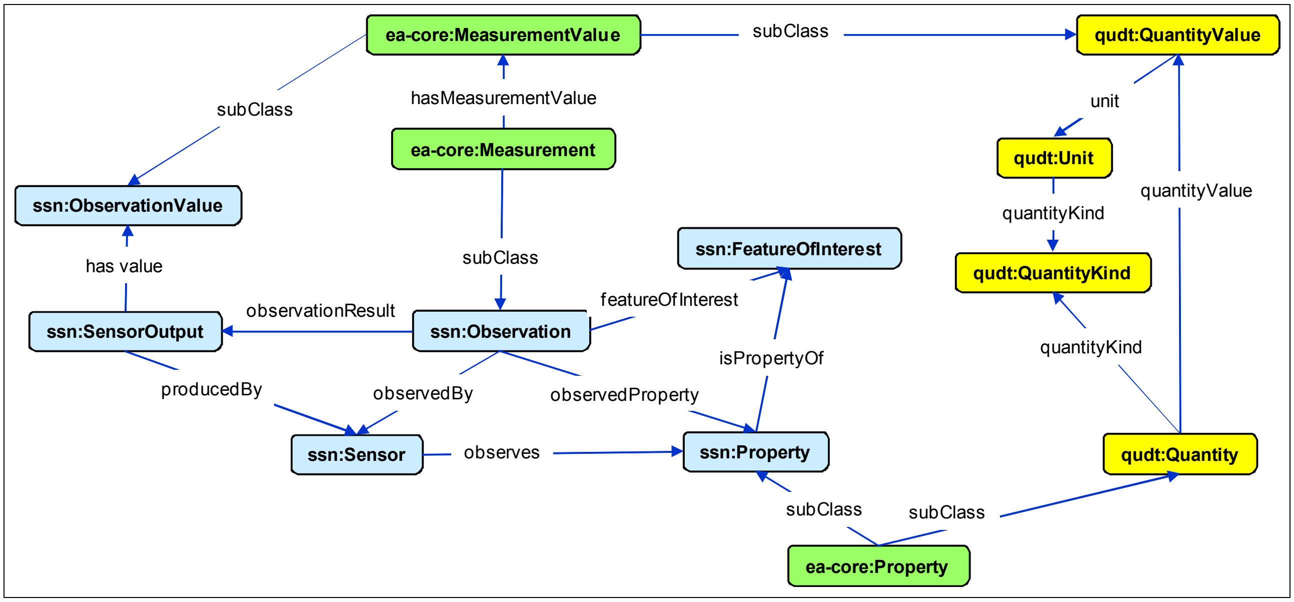

Figure 4   IEM Ontology Meta-model - observation and measurement

Here are the classes for observations and measurement:

- ea-core:Property: This class is a subclass of ssn:Property and qudt:Quantity. The ea-core property class links the ssn:Property and the qudt:Quantity concepts because they are equivalent. The qudt:Quality is tied to *Quantity Kind* and *Unit* in QUDT. All EA properties, such as water temperature, water current speed, water current direction, and sedimentation levels, are instances of ea-core:Property.

- ssn:Property:

  This is an observable quality of an event or object. It is not a quality of an abstract entity, which is allowed by DUL's Quality, but rather an aspect of an entity that is intrinsic to and cannot exist without the entity, and is observable by a sensor. For more information, see the ssn#Property definition, found at:

  `http://www.w3.org/2005/Incubator/ssn/ssnx/ssn#Property`

- qudt:Quantity:

  A *quantity* is an observable property of an object, event, or system that can be measured and quantified numerically. Quantities are differentiated by two attributes that make up the essential parameters that are needed to formalize the structure of quantities: kind and magnitude. The kind attribute of a quantity identifies the observable property that is quantified, for example, length, force, and frequency and the magnitude of the quantity expresses its relative size compared to other quantities of the same kind. For more information, see the qudt:Quantity definition, found at:

  `http://www.qudt.org/qudt/owl/1.0.0/qudt/index.html#Quantity`

Example 3 shows an example of an ea-core:Property.

Example 3 ea-core:Property

```
ea-core:Property
        a                  owl:Class ;
        rdfs:comment       "All EA properties like water temperature, water current speed, water
                           current direction, sedimentation levels, etc. are instances of
                           ea-core:Property.”^^xsd:string ;
        rdfs:label         "Property"^^xsd:string ;
        rdfs:seeAlso       <http://www.w3.org/2005/Incubator/ssn/ssnx/ssn#Property> ;
        rdfs:subClassOf    ssn:Property;
        rdfs:subClassOf    qudt:Quantity .
```

- ea-core:Measurement: This class is a subclass of ssn:Observation. It is linked to the ea-core:MeasurementValue entity through the ea-core:hasMeasurementValue property, and can include additional EA attributes that are specific to measurement but are not included in ssn:Observation.
- ea-core:MeasurementValue: This class is a subclass of the ssn:ObservationValue and links ssn:ObservationValue with qudt:QuantityValue (which is associated with units of measure (qudt:Unit) and quantity kinds). The value of the observation in SSN is expressed as an instance of the class ssn:ObservationValue. However, SSN does not restrict the format of an observation value; the properties can be defined by the user or imported from a third-party ontology. This scenario defines observation values as qudt:QuantityValue through the ea-core:MeasurementValue class. As the SSN ontology is domain independent and leaves the observed domain unspecified, you must extend ssn:ObservationValue with a class that links it to quantity kinds and units of measure in QUDT.

Example: Sedimentation level measurements for a coral reef

Figure 5 shows a way to model the sedimentation level measurement for a coral reef. Coral_558 is an ea:CoralReef, which has the property ea:AccumulatedSediment, which is a property with the qudt:Length quantity kind. The measurement (ea:Coral_558SedimentationLevel_M) has Coral_558 as its feature of interest, and AccumulatedSediment as its observed property. The ea:Coral_558SedimentationLevel_M measurement is associated with a measurement value (ea:Coral_558SedimentationLevel_MV) with links to the numeric value and unit of measure.

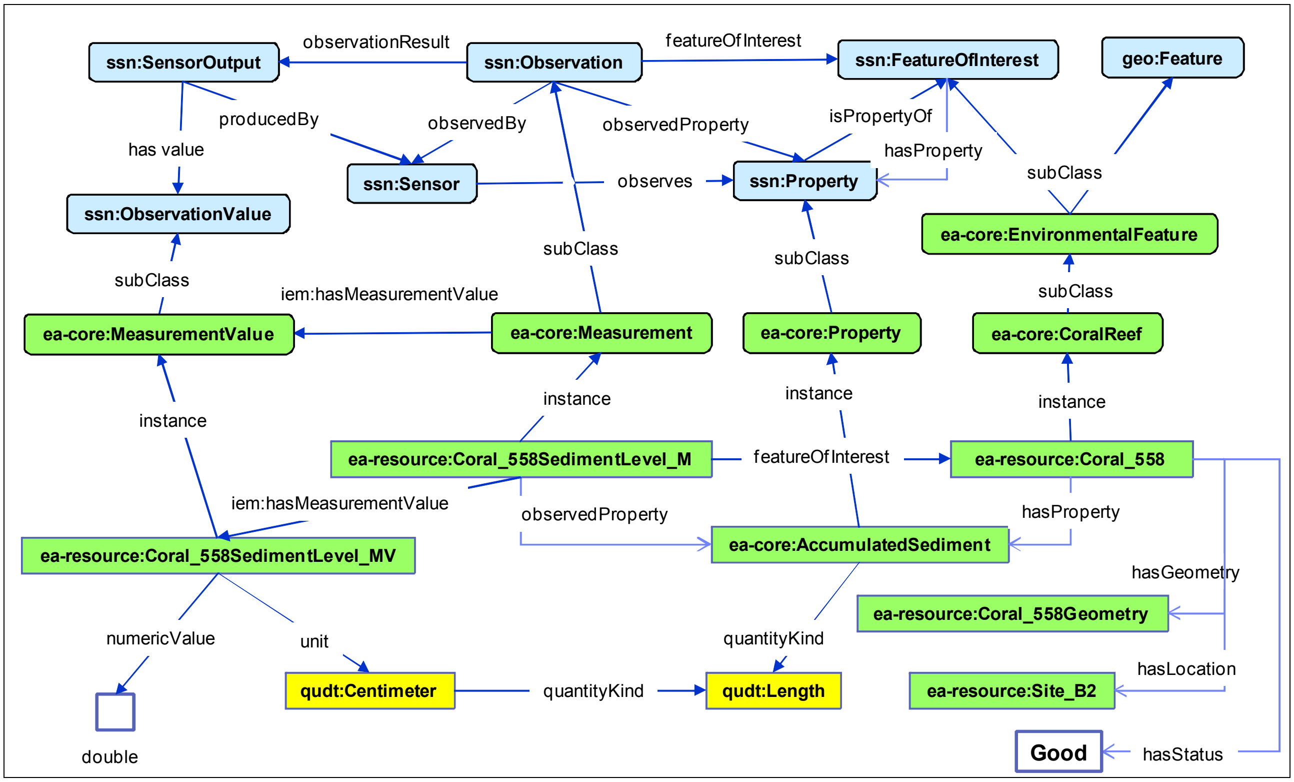

Figure 5 Ontology Meta-model - observation and measurement

The rounded boxes in the figure are class entities, and the square boxes are individual entities that are instances of the class. The green boxes are elements of the EA domain ontology.

Figure 5 models ea:AccumulatedSediment as an individual instance of the class ea:Property. It is also possible to model ea:AccumulatedSediment as a subclass of ea:Property.

Sensors, sensing systems, and platforms

Various types of sensors are used in EA that measure different parameters and produce measurement data of different data types. For example, the Recording Doppler Current Profiler (RDCP) sensor enables the creation of a water current profile by measuring water current speed and direction. Another sensor example is a camera with light that provides visual observations in the form of video and still pictures.

All sensors in EA are modeled as instances of *ssn:Sensor*. A sensor implements sensing, that is, a sensor is any entity that can follow a sensing method and thus observe some property of a feature of interest. Sensors can be physical devices, computational methods, a laboratory setup with a person following a method, or any other thing that can follow a sensing method to observe a property.

Figure 6 shows an example of modeling a water current sensor, where sensor WCS-01 is an instance of an ea:CurrentSensor. WCS-01 observes two properties: ea:WaterCurrentSpeed, and ea:WaterCurrentDirection.

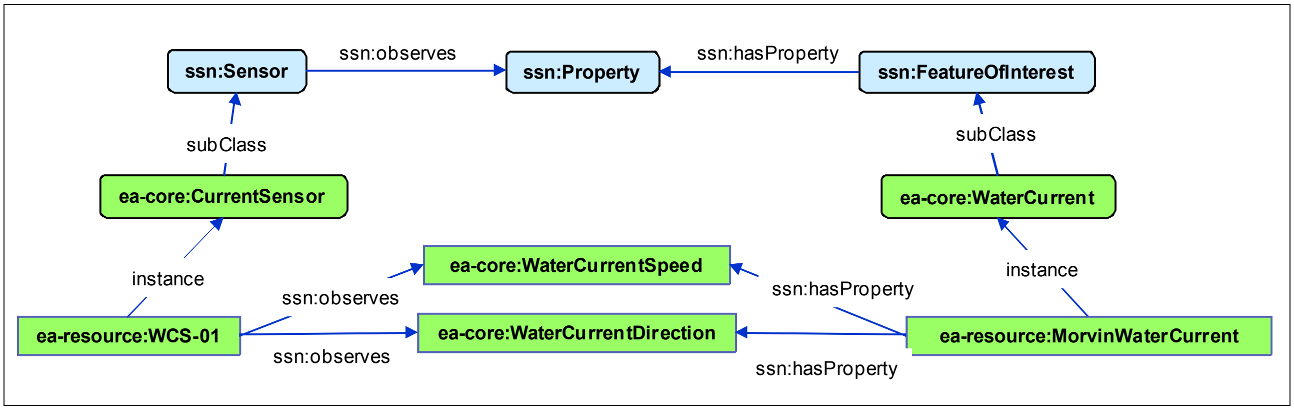

Figure 6 All sensors in EA are modeled as instances of ssn:Sensor

All sensor instances can specify the following properties:

- *ssn:observes* specifies the property that the sensor observes, such as water current speed.
- *ssn:detects* specifies the stimulus that the sensor detects, for example, ultraviolet radiation.
- *ssn:onPlatform* specifies the platform that the sensor is attached to, for example, Lander-01.
- *dul:hasLocation* specifies the site where the sensor is deployed, for example, Deepwater_Site_B2 (site B2 within a larger site called Deepwater).

The actual geo-spatial location of the sensor is specified through the location of the platform to which the sensor is attached, such as the Lander. For example, we model the Lander as a ssn:Platform and as a geo:Feature. The geo:Feature gives the Lander a geometry that can identify its location.

Modeling sensor systems with multiple sensors

Many sensor systems have multiple sensors that are attached to the sensor platform. For example, the Recording Doppler Current Profiler (RDCP) measures the current speed and direction, and is a system with the option to include different sensors that can measure different water properties, including Temperature, Conductivity, Pressure, Oxygen, Turbidity (number of particles in water), and Fluorescence. For more information, see the following website:

`http://www.aanderaa.com/productsdetail.php?Recording-Doppler-Current-Profiler-RDCP-33`

There are different ways to model this sensor system. One of these ways is shown in Figure 7.

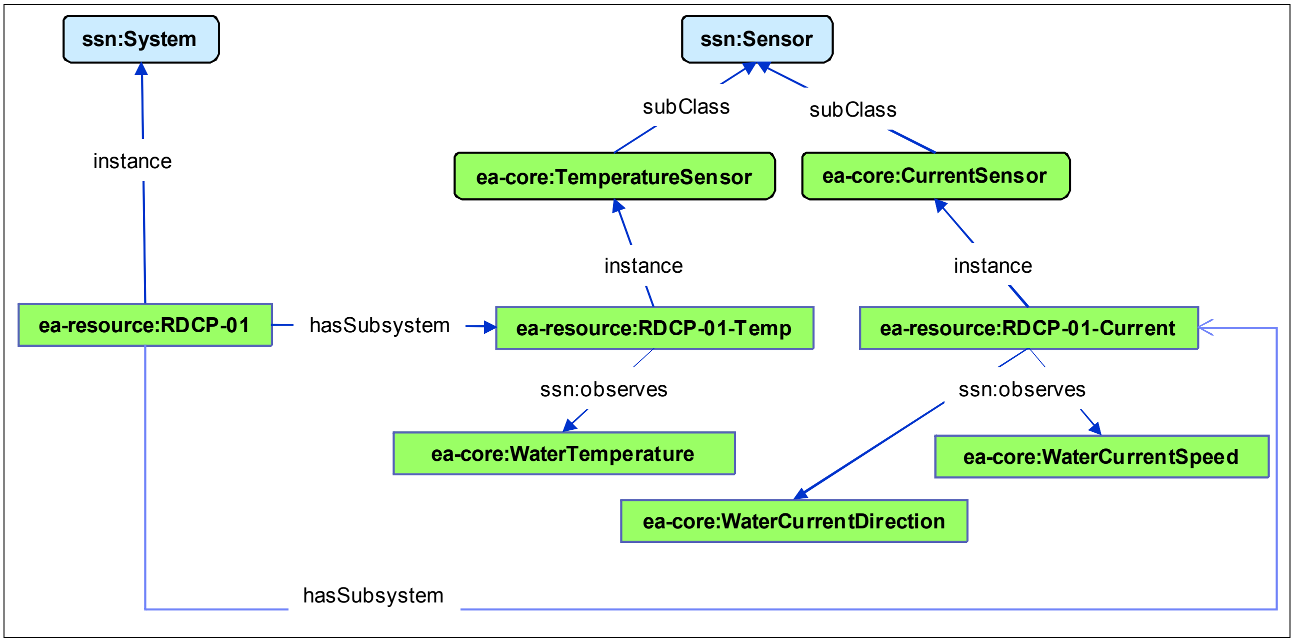

Figure 7 RDCP-01 (an instance of an RDCP system) as an instance of a ssn:System

Figure 7 shows RDCP-01 (an instance of an RDCP system) as an instance of a ssn:System with two subsystems:

- RDCP-01-Temp (an instance of ea:TemperatureSensor), which observes the water temperature property
- RDCP-01-Current (an instance of ea:CurrentSensor), which observes the current speed and direction

Modeling the Lander: A sensor platform

The EA Lander is a platform that includes sensing equipment and supporting infrastructure. The Lander is deployed in a specific location on the sea bed where certain environmental measurements (such as water temperature and water current speed and direction) are wanted. This example models the Lander as a Semantic Sensor Network (SSN) platform to which the sensors are attached. In addition, to represent the location and associated coordinates, the Lander is modeled as a geographic feature using the Geo-SPARQL ontology.

The Lander is a ssn:Platform and a geo:Feature, as shown in Figure 8 and Example 4 on page 19.

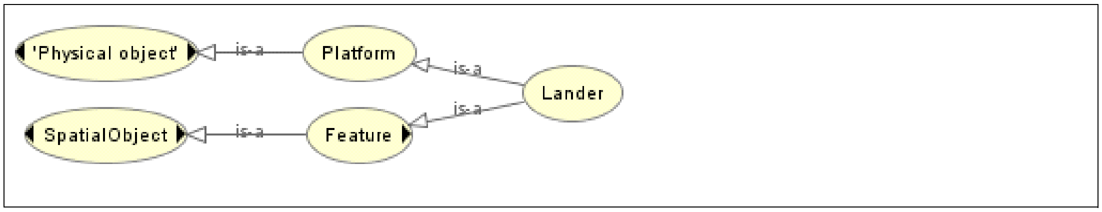

Figure 8 Modeling the Lander

Example 4 Modeling the Lander

```
xmlns:ssn="http://purl.oclc.org/NET/ssnx/ssn#"
xmlns:geo="http://www.opengis.net/ont/geosparql#"

ea:Lander     a              owl:Class ;
              rdfs:label     "Sub-sea Lander"^^xsd:string ;
              rdfs:seeAlso   <http://www.w3.org/2005/Incubator/ssn/ssnx/ssn#Platform> ;
              rdfs:subClassOf ssn:Platform;
              rdfs:subClassOf geo:Feature .
```

ssn:Platform is a platform that is defined in the Semantic Sensor Network (SSN) ontology as an entity to which other entities can be attached, such as sensors and other platforms. For example, a post might act as the platform, a buoy might act as a platform, or a fish might act as a platform for an attached sensor. For more information, see the ssn:Platform definition, found at:

`http://www.w3.org/2005/Incubator/ssn/ssnx/ssn#term-ssn:Platform`

When you create an instance of the Lander class, you can use the inherited Lander properties to attach other entities to the Lander to pinpoint the geo-spatial location of the Lander, and to determine its location relative to a specified site. For example, if you add the statement “`ssn:attachedSystemea:WCS-01;`” to the definition of a Lander Instance (for example, Lander-1), you indicate that Lander-1 has the ea:WCS-01 system attached to it. In this example, the WCS-01 is a water current sensor, which is an instance of a ssn:Sensor. The Lander instance can specify the following items:

- geo:hasGeometryea:Lander-1-Geometry: Specifies the Lander's location.
- dul:hasLocationea:Deepwater_Site_S1: Specifies that the Lander is in the Deepwater site S1.

Modeling a site and enterprise

For modeling a site and enterprise, this example, uses the ea-core:Site class. This class is a subclass of geo:Feature. An example is Deepwater site. Figure 9 shows an example of ea-core:Site.

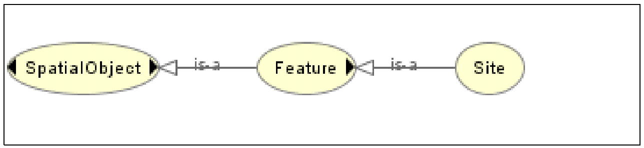

Figure 9 ea-core:Site

A site should be modeled to include other sites or points or areas of interest. For example, the Deepwater site can include multiple sites (drilling locations or areas) within it, such as Site_A1, Site_A2, and Site_B1.

The *ea-core:Enterprise* class is a subclass of dul:Organization. An example is Statoil. You might need to represent the enterprises that own and operate at different sites. Figure 10 shows an example of ea-core:Enterprise.

Figure 10 ea-core:Enterprise

In this example, you want to assert that an enterprise *registers* a site and a site *isRegisteredby* by an enterprise, as shown in Example 5.

Example 5 An enterprise registers a site and a site isRegisteredby by an enterprise

```
ea-core:Enterprise     ea-core:registers        ea-core:Site
ea-core:Site           ea-core:isRegisteredby   ea-core:enterprise
```

Environmental Analytics modeling example

Consider the example of modeling a Lander that includes a water current sensor that measures the water current speed and direction, and a temperature sensor that measures the water temperature at the Lander site. In this example, consider the following items:

- Lander-1 is a Lander platform that has the following two sensors that are attached to it:
  - WCS-01 is a water current sensor that can measure the water current speed and direction.
  - TS-01 is a temperature sensor that can measure the water temperature.
- Lander-1 is deployed on the sea bed at location (point) latitude = 65.1566, longitude = 6.4854, and altitude = -353.

Modeling the Lander

The EA Lander is a platform that includes sensing equipment and supporting infrastructure. The Lander is deployed in a specific location on the sea bed where certain environmental measurements (such as water temperature and water current speed and direction) are wanted. This example models the Lander as a Semantic Sensor Network (SSN) platform to which the sensors are attached. In addition, to represent the location and associated coordinates, the Lander is modeled as a geographic feature using the Geo-SPARQL ontology.

Example 6 shows an example of a modeled Lander.

Example 6 Modeling the Lander

```
Lander is a ssn:Platform; Lander is a geo:Feature;

xmlns:ssn="http://purl.oclc.org/NET/ssnx/ssn#"
xmlns:geo="http://www.opengis.net/ont/geosparql#"

ea:Lander    a               owl:Class ;
             rdfs:label      "Sub-sea Lander"^^xsd:string ;
             rdfs:seeAlso    <http://...> ;
             rdfs:subClassOf ssn:Platform;
             rdfs:subClassOf geo:Feature .
```

Geo:Feature: There are three key classes in the GeoSPARQL ontology:

- geo:Feature: A thing that can have a spatial location, for example, a park or a monument
- geo:Geometry: A representation of a spatial location, for example, a set of coordinates
- geo:SpatialObject: A superclass of both features and geometries

Both geo:Feature and geo:Geometry are sub-classes of geo:SpatialObject. The geo:hasGeometry property links features (a thing) to their geometry (their location). By separating the entities and their locations, GeoSPARQL allows multiple geometries to be linked to a feature for various purposes.

This example links the EA domain ontology to the GeoSPARQL ontology by making the Lander class in the EA domain ontology a subclass of geo:Feature, meaning that instances of the Lander class can point to a geo:Geometry with the geo:hasGeometry property.

Modeling Lander-1: The Lander instance

Lander-1 is an instance of an ea:Lander that has sensors WCS-01 and TS-01 attached to it, and is deployed at location with latitude = 65.1566, longitude = 6.4854, and altitude = -353, as shown in Example 7.

Example 7 The Lander instance

```
ea:Lander-1     a                   ea:Lander ;
                rdfs:label          "Sub-sea Lander-1 - deployed at the Deepwater field" ;
                ssn:attachedSystem  ea:WCS-01;
                ssn:attachedSystem  ea:TS-01;
                geo:hasGeometry     ea:Lander-1-Geometry;
                dul:hasLocation     ea:Deepwater_Site_S1 .
```

Modeling Lander-1-Geometry: A geo:Point instance

The Lander-1 geo-spatial location is described as a geo:Point and specified as shown in Example 8.

Example 8 A geo:Point instance

```
ea:Lander-1-Geometry       a          geo:Point ;
                           rdfs:label "Lander-1 point location" ;
                           geo:asWKT  "POINT(65.1566  6.4854  -353)" ^^geo-sf:WktLiteral .
```

Modeling WCS-01 Sensor: The Water Current and Speed Sensor instance

The WCS-01 sensor is two sensors in one: one measures the current speed, and the other one measures the current direction.

Additional information for the Observation and Measurement ontology

The O&M model is specified in the OGC standards (for more information, see http://www.opengeospatial.org/). The model, which is shown in Figure 11, is based on the observation-feature-property pattern, which is useful for capturing the metadata that is associated with the estimation of feature properties. An *observation* is an action with a result that has a value describing some phenomenon. An observation binds a result to a feature of interest, upon which the observation was made. The observed property is a property of the feature of interest. An observation uses a procedure to determine the value of the result, which can involve a sensor or observer, analytical procedure, simulation, or other numerical process.

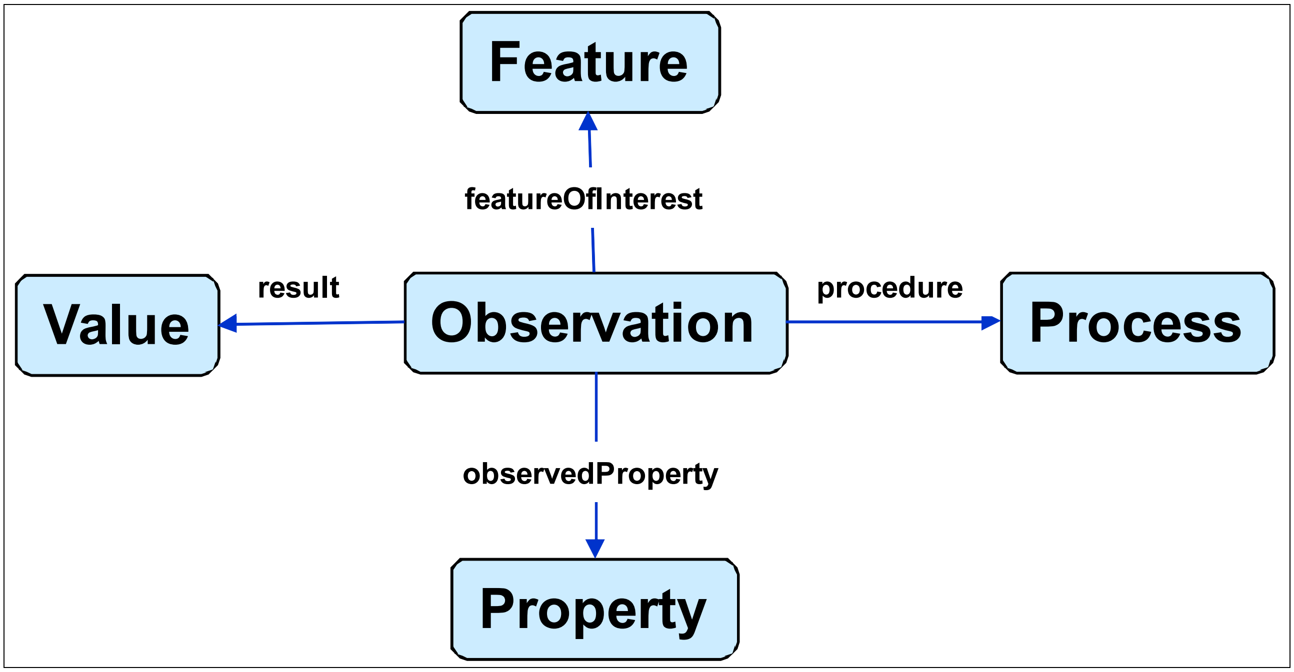

Figure 11 The Observation and Measurement model

In Figure 11:

- *Observation* is an act of observing a property or phenomenon, with the goal of producing an estimate of the value of the property. It is specialized event whose result is a data value.
- *Property* is a characteristic of a feature. The observed property is a property of the feature of interest. A facet or attribute of an object is referenced by a name (what is measured).
- *Feature* is an abstraction of real-world phenomena. It is the target object of the observation.
- *Result* is an estimate of the value of some property that is generated by a known procedure.
- *Procedure* is a method, algorithm, or instrument, or a system of these items, which can be used in making an observation.

Observation and Measurement model in SSN

The observation model in SSN ontology (see Figure 12) has its roots in the OGC O&M data model. It revolves around the Stimulus-Sensor-Observation pattern where sensors observe properties of features of interest by detecting stimuli, for example, changes in the physical world (directly or indirectly) that are related to these properties, and transforming them to another representation as results. Sensors implement a procedure that describes the transformation of stimuli to results. Observations are the contexts that bring sensors and stimuli together. They are described by procedures that determine how a certain observation must be carried out.

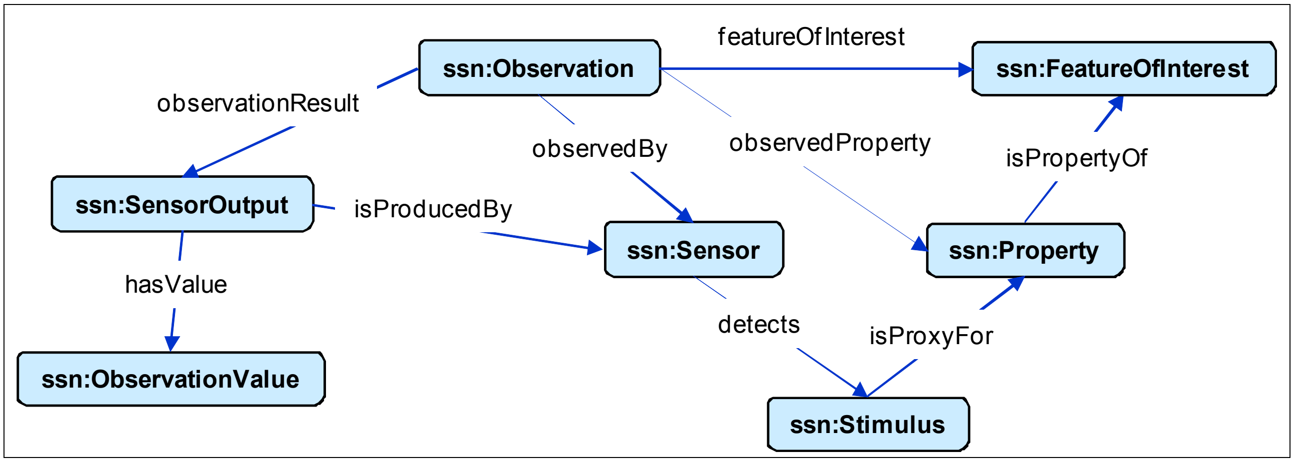

Figure 12 The observation model in the SSN ontology

Additional information for the quantity kinds, quantity values, and units of measure ontologies

The concepts that are modeled in the quantity kinds, quantity values, and units of measure ontologies are shown in Figure 13.

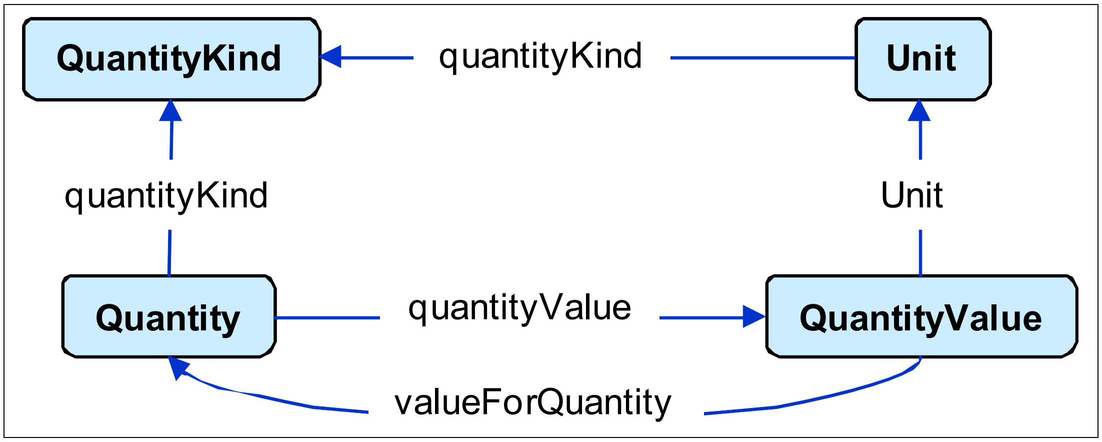

Figure 13 Quantity kinds, quantity values, and units of measure ontologies

In Figure 13 on page 23:

- *Quantity Kind* is any observable property that can be measured and quantified numerically. Examples include physical properties Line Length, Mass, Time, and Force. Other properties can include Currency, Interest Rate, and Price to Earning Ratio.
- *Quantity* is an observable property of an object, event, or system that can be measured and quantified numerically. Examples include the mass of a hydrogen atom, the temperature at a certain site, or the duration of a specific meeting. The attributes include the following ones:
  - *Kind* identifies the observable property that is quantified.
  - *Magnitude* expresses its relative size compared to other quantities of same kind.
- *Unit of Measure* is a particular quantity of a given kind that is chosen as a scale for measuring other quantities of the same kind. Examples include Meters, Kilograms, and Volts.
- *Quantity Value* is the numerical value of a quantity's magnitude with respect to a chosen unit of measure for the corresponding quantity kind. Examples include “5 kilograms” or “3 meters”.

References and related work

Here are the references and related work for this solution:

- *Ontology*, found at:

  http://tomgruber.org/writing/ontology-definition-2007.htm
- *Resource description framework application development in DB2 10 for Linux, UNIX, and Windows, Part 1: RDF store creation and maintenance*, found at:

  http://www.ibm.com/developerworks/data/tutorials/dm-1205rdfdb210/
- Open Services for Lifecycle Collaboration (OSLC):

  http://open-services.net/
- IBM Watson™ Solutions:

  http://www-03.ibm.com/innovation/us/watson/index.shtml
- *Making the Argument for Semantic Technologies*, found at:

  http://www.mkbergman.com/974/making-the-argument-for-semantic-technologies/
- QUDT - Quantities, Units, Dimensions, and Data Types Ontologies:

  http://www.qudt.org
- Open Geo-spatial Consortium (OGC):

  http://www.opengeospatial.org/
- *Observation and Measurement - Part 1 - Observation schema*, found at:

  https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CC0QFjAA&url=http%3A%2F%2Fportal.opengeospatial.org%2Ffiles%2F%3Fartifact_id%3D22466&ei=qGy4UfmvPIOG9QTY44DICA&usg=AFQjCNHRMOMN4FRklzFBHYlDy47TU4ZgtQ&sig2=-ysewv-LECutX3UDNtVc6A
- Semantic Sensor Network (SSN) Ontology:

  http://www.w3.org/2005/Incubator/ssn/ssnx/ssn

- GeoSPARQL Ontology:

  `http://www.opengeospatial.org/standards/geosparql`
- W3C Basic Geo (WGS84 lat/long) vocabulary:

  `http://www.w3.org/2003/01/geo/`
- *The role of semantic models in smarter industrial operations*, found at:

  `http://www.ibm.com/developerworks/xml/library/x-ind-semanticmodels/index.html`
- ssn:FeatureOfInterest definition:

  `http://www.w3.org/2005/Incubator/ssn/ssnx/ssn#term-featureOfInterest`
- ssn:Platform definition:

  `http://www.w3.org/2005/Incubator/ssn/ssnx/ssn#term-ssn:Platform`
- ssn#Property definition:

  `http://www.w3.org/2005/Incubator/ssn/ssnx/ssn#Property`
- qudt:Quantity definition:

  `http://www.qudt.org/qudt/owl/1.0.0/qudt/index.html#Quantity`
- IBM Redbooks® Solution Guide *Smarter Environmental Analytics Solutions: Offshore Oil and Gas Installations example*, TIPS1131

Author

This paper was produced by Dr. Akram Bou-Ghannam.

Dr. Akram Bou-Ghannam is an executive IT architect with IBM Software Group Industry Solutions, defining cross-industry solutions for enabling operational business intelligence to improve operational decision making and manage and mitigate operational risk. He is an advocate of semantic technology, semantic integration, and contextual computing. He has deep experience in defining industry solutions for the optimization of the operations and maintenance of high-value engineering assets that are used and deployed in various industries, including Oil & Gas, Mining & Construction, Aerospace & Defense, Energy & Utilities, and so on. He is also knowledgeable about various associated industry standards. Dr. Bou-Ghannam is an IBM Master inventor and an accomplished author. He holds a Ph.D. degree in Electrical Engineering from the University of Florida, Gainesville (1990).

Thanks to the following people for their contributions to this project:

Jone Aarre, Frode Myren, Oystein Haaland
IBM Norway

David Noller and Paul Peters
IBM US

Peter G. W. Kluge
IBM Germany

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

`ibm.com/redbooks/residencies.html`

Stay connected to IBM Redbooks

- Find us on Facebook:

  `http://www.facebook.com/IBMRedbooks`
- Follow us on Twitter:

  `http://twitter.com/ibmredbooks`
- Look for us on LinkedIn:

  `http://www.linkedin.com/groups?home=&gid=2130806`
- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  `https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm`
- Stay current on recent Redbooks publications with RSS Feeds:

  `http://www.redbooks.ibm.com/rss.html`

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document REDP-5081-00 was created or updated on January 31, 2014.

Send us your comments in one of the following ways:
- Use the online **Contact us** review Redbooks form found at:
  ibm.com/redbooks
- Send your comments in an email to:
  redbooks@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

DB2®
IBM®
IBM Watson™
Rational®
Redbooks®
Redpaper™
Redbooks (logo) ®

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.