PhUSE 2013

Paper DH03

Semantic Technology and CDISC Standards

Frederik Malfait, IMOS Consulting, Switzerland
Scott Bahlavooni, Independent, Boston, USA

ABSTRACT

Over the past few years there has been increased interest in semantic technology as a foundation for metadata registries (MDR) in general and clinical data standards in particular. In this paper we will introduce the principles of RDF and Linked Data, and show how they can be applied to metadata management. We also highlight its role in a number of projects and initiatives, such as the Roche MDR for clinical data standards, the FDA/PhUSE Semantic Technology project, and a number of public initiatives. Of particular interest may be the use of RDF to represent CDISC standards such as SDTM and how RDF is well positioned to represent and link different types of metadata related to domains, variables, controlled terminology, and validation checks.

INTRODUCTION

Most attendees of the Pharmaceutical Users Software Exchange will be very familiar with CDISC standards, some will have a keen interest in the definition and use metadata as a means to maximize the benefit of implementing data standards, but we expect not very many to have a working knowledge of semantic technology. In this paper we explain the principles of the Resource Description Framework (RDF) and related W3C semantic standards, and show how they are a natural fit for representing data and metadata within a single framework. We also show how the ISO 11179 standard for Metadata Registries (MDR) can be integrated into this framework to provide a solid foundation for managing corporate CDISC based clinical and non-clinical trial data standards.

W3C SEMANTIC STANDARDS

The W3C has defined semantic web standards with the goal of representing resources on the web as linked data rather than linked pages. This representation has two primary advantages. First, it makes information on the web explicit and computable, rather than hidden in HTML pages. Second, it provides a mechanism to reuse and link computable data, rather than creating “dumb” links between pages. The fundamental language of these standards is that of a formal language, in mathematical terms a first-order predicate language. Beyond or outside the web it can be used to create ontologies (formal knowledge models) about almost any kind of subject. In this capacity, the W3C semantic web standards have started to find their way as a knowledge representation language in corporate information management. We have used these languages to create computable, machine-readable models of CDISC-based clinical trial data standards at Hoffmann-La Roche.

THE RESOURCE DESCRIPTION FRAMEWORK (RDF)

RDF is the foundation of the W3C semantic standards. It is a language used to describe any resource, not just resources on the web. For example, all of the following can be described as resources: the SDTM Adverse Event (AE) domain, the AE domain variable Adverse Event Outcome (AEOUT), or the controlled terminology list C66768 describing the possible values of AEOUT. The W3C requires RDF resources to be identified by a Uniform Resource Identifier (URI). A URI usually takes a format similar to that of a URL with the distinction that a URI is not necessarily required to be reachable as a network resource. The following are examples of tentative URIs of CDISC resources:

```
http://rdf.cdisc.org/sdtmig-3-1-2/Table.AE
http://rdf.cdisc.org/sdtmig-3-1-2/Column.AE.AEOUT
http://rdf.cdisc.org/sdtm/terminology/C66768
```

As seen above, these URI values tend to be long, so RDF has a way to shorten them using namespaces. For example, assume the prefix `sdtmig` stands for `http://rdf.cdisc.org/sdtmig-3-1-2/`, and the prefix `sdtmt` stands for `http://rdf.cdisc.org/sdtm/terminology/`, then the same URIs can be represented as follows:

```
sdtmig:Table.AE
sdtmig:Column.AE.AEOUT
sdtmt:C66768
```

In addition to naming the resources, we want to say things about these resources, i.e. describe these resources. In RDF, descriptions or statements are in the form of subject/predicate/object sentences. Subjects are what is being

described, predicates are the subject properties or relationships, and objects are either literal property values or other related resources.

The following are example statements. The predicates have been underlined for clarity:

- AEOUT has label “Adverse Event Outcome”
- AEOUT is in the domain AE
- AE has domain label “Adverse Events”
- AEOUT has code list C66768
- C66768 is extensible “true”

The object in the first, third, and fifth statement is a literal. The first two are string literals and the last one is a boolean literal. The objects in the other statements are resources. In this example, the resource AE is an object of the second statement and a subject of the third statement. Each predicate in RDF is itself a resource with its own URI. As such, the previous statements can formally be expressed in RDF as follows.

sdtmig:Column.AE.AEOUT	mms:dataElementLabel	[2]Adverse Event Outcome”
sdtmig:Column.AE.AEOUT	mms:dataElementDomain	sdtmig:Table.AE
sdtmig:Table.AE	mms:domainLabel	“Adverse Events”
sdtmig:Column.AE.AEOUT	sdtmigs:dataElementCodelist	sdtmt:C66768
sdtmt:C66768	sdtmts:isExtensibleCodelist	“true”

Please note, how each statement has three components: a subject represented by a URI, a predicate represented by a URI, and an object represented by a literal or a URI. Such RDF statements are called triples.

GRAPHS AND LINKED DATA

The previous example may lead to the conclusion that RDF seems almost too simple to be useful; however, the real power of RDF data becomes clear when viewed, not as a list of triples, but as a network of resources connected by predicates. The following example shows the resources described above, with an additional resource for each codelist element.

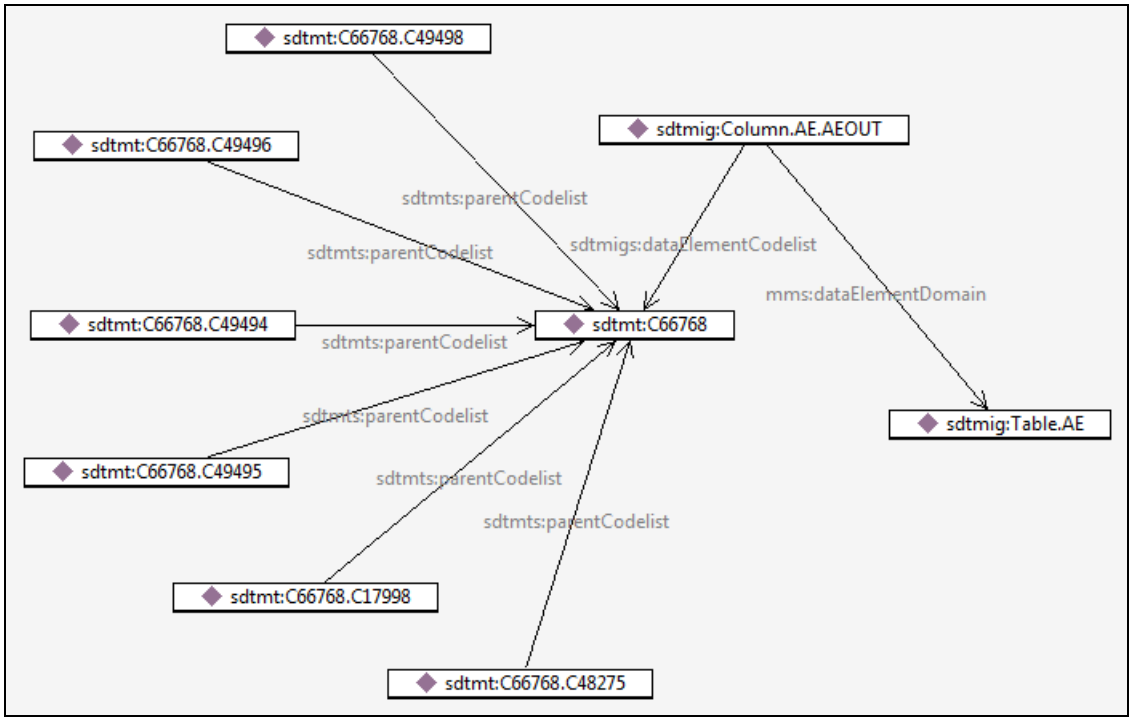

Figure 1. Sample RDF Graph

Simlar to hyperlinked web pages, resources can be linked using predicates to build a network or a graph of interconnected resources. As a data representation model, graphs are more flexible than tables, easier to expand over time with new content (additive), and easier to federate across the network. At Hoffmann-La Roche we currently have represented the complete CDISC-aligned data collection and data tabulation standards in RDF using approximately 60’000 resources and 400’000 triples.

RDF is extremely scalable. For example, dbpedia.org is an RDF knowledge base that extracts structured information from Wikipedia and currently hosts close to two billion triples. Resources are linked together to create graphs, but things get more interesting when resources are linked across graphs, e.g. dbpedia.org has about 30 million triples linking resources to external RDF data sets on the web. This web based network of RDF data sets is known as the Linked Open Data (LOD) cloud and looks as follows (as of 2011).

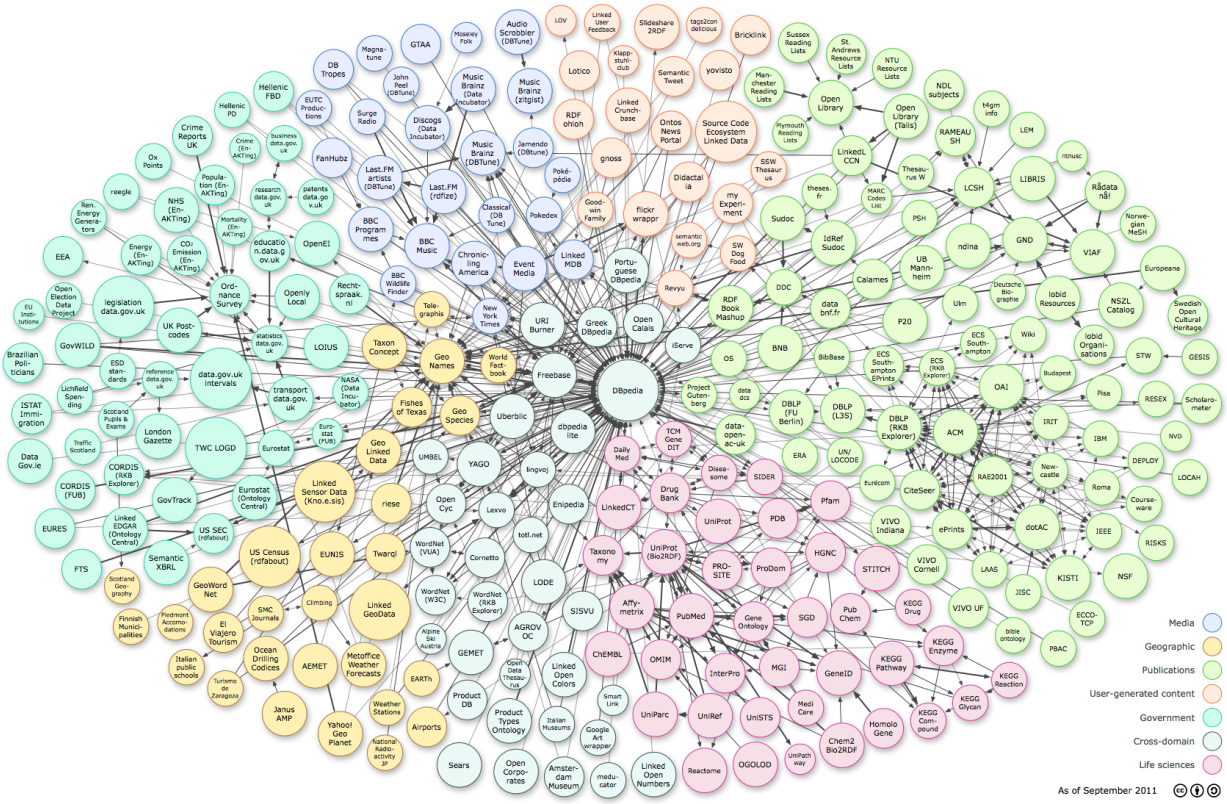

Figure 2. Linked Open Data cloud diagram by Richard Cyganiak and Anja Jentzsch at lod-cloud.net

The life sciences are in red at the bottom center. A plethora of data is available in RDF simply waiting to be linked. Additional RDF data sources become available on a continuous basis. Another interesting source is the National Center for Biomedical Ontology at bioontology.org. The BioPortal on this site offers browsing and federated search across about 360 biomedical ontologies. Hoffmann-La Roche links the data standards to the RDF version of the NCI Thesaurus for term lookup and plans to link the biomarker data standards to the HGNC as a source of sponsor-defined controlled terminology for gene locations.

RDF SCHEMA (RDFS) AND WEB ONTOLOGY LANGUAGE (OWL)

RDF provides the foundation to name, describe, and link things, but is limited in defining more structural information. To this end, the W3C has created additional vocabularies to create schemas (RDFS) and full-fledged class based ontologies (OWL). RDFS and OWL are themselves written in RDF.

RDFS and OWL enable inference making, i.e. deriving new triples from existing triples. Consider the following statements:

- `sdtmig:Column.AE.AEOUT  mms:dataElementDomain  sdtmig:Table.AE`
- `mms:hasDataElement  owl:inverseOf  mms:dataElementDomain`

An OWL compliant inference engine will be capable of deriving a new statement based on the OWL defined meaning of the owl:inverseOf predicate:

- `sdtmig:Table.AE  mms:hasDataElement  sdtmig:Column.AE.AEOUT`

OWL defines a long list of predicates that are defined in terms of derivations an inference engine can make when data is described in an OWL ontology. OWL enables the definition of predicates that are symmetric, transitive, functional, inverse functional, and much more.

Many people have been using OWL to create extensive ontologies where inference making is a key part of the modeling effort. More recently, a modest trend can be seen in corporate modeling where a targeted subset of OWL

is used to create RDF based object models, i.e. RDFS and OWL are used to create a lean class based model for a certain domain rather than trying to package a lot of knowledge for inference making. OWL classes and RDFS domain/range information are integral to this approach. The following example illustrates the key ideas.

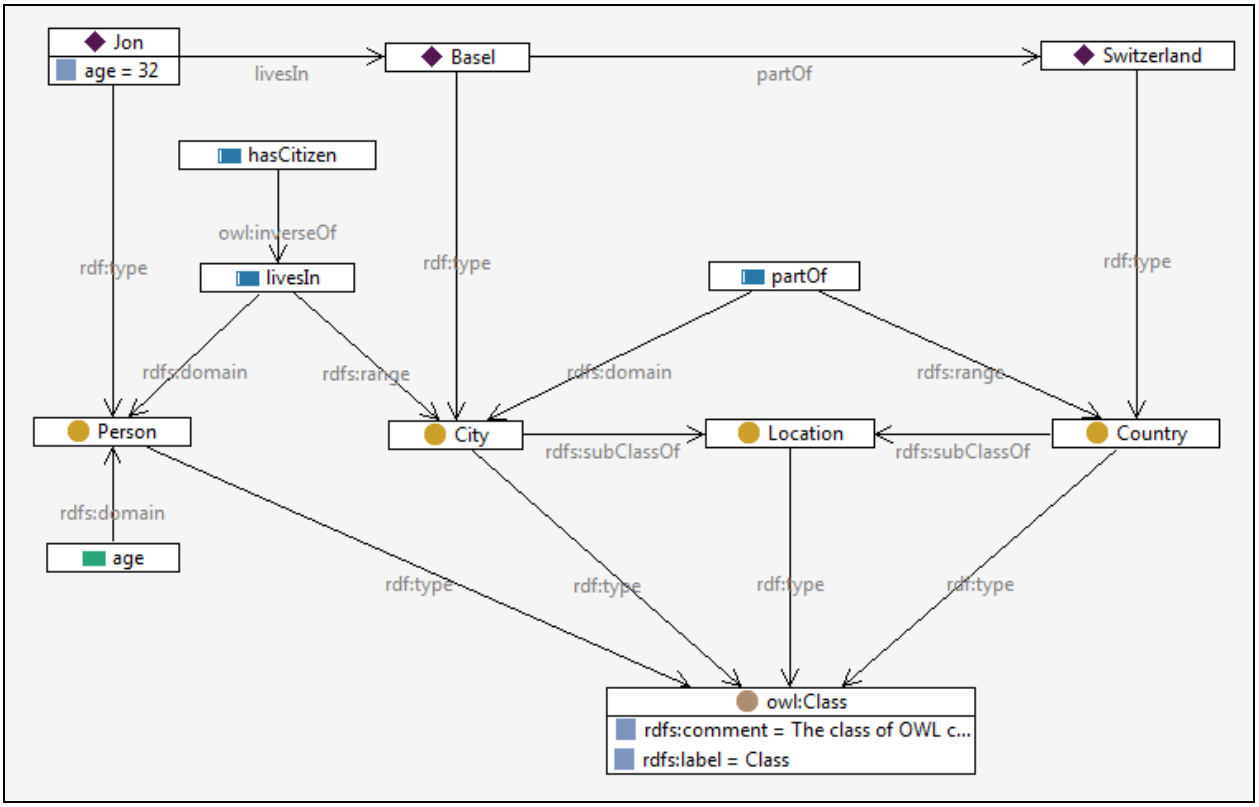

Figure 3. OWL Class Based Modeling

Only a small subset of RDFS and OWL is used here. After creating the classes Person and Location (as subjects of rdf:type owl:Class), resources can be made instances of classes, e.g. Basel rdf:type City. Additionally, City and Country are made subclasses of Location using rdfs:subClassOf. Applying inference makes both City and Country classes as well. These tools are already enough to create class based models and organize classes in hierarchies. Consider now the predicates, e.g. livesIn always goes from a Person to a City. This can be expressed by saying livesIn rdfs:domain Person and livesIn rdfs:range City.

Types, domains, and ranges are fundamental tools to describe the structure of the data, i.e. they enable the creation of class-based schemas to model the data. Schemas describe data and are themselves expressed in RDF. As such, they can be integrated with the RDF data sets they describe.

The following example describes part of a schema to express SDTM.

```
·  mms:DataElement          rdf:type      owl:Class
·  sdtmig:Column.AE.AEOUT   rdf:type      mms:DataElement
·  mms:dataElementLabel     rdf:type      owl:DataTypeProperty
·  mms:dataElementLabel     rdfs:domain   mms:DataElement
·  mms:dataElementLabel     rdfs:range    xsd:string
```

Previous examples introduced the Adverse Event Outcome resource to represent a data element from the SDTM AE domain. The additional statements turn Data Element into a class and Adverse Event Outcome into an instance of that class. Further, data element label is a predicate that defines a data type for Data Element (domain) and the actual type is xsd:string (range). Data type properties can be any XML schema data type.

SIMPLE KNOWLEDGE ORGANIZATION SYSTEM (SKOS)

Many applications of knowledge representation deal with defining terms as concepts and relating concepts on the basis of broader or narrower relationships in a very simple and straightforward way. The W3C has defined a small RDF vocabulary called SKOS for this purpose.

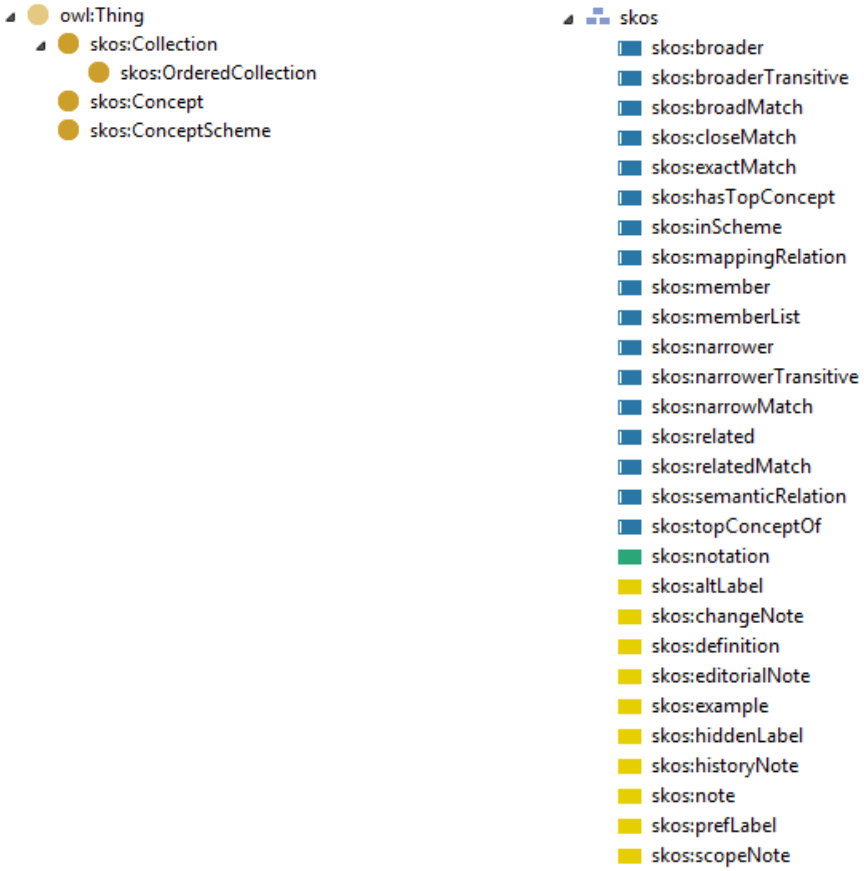

Figure 4. SKOS Classes and Predicates

Hoffmann-La Roche has used this vocabulary to link data elements to concepts, which are further linked to terms in the NCI Thesaurus. Hoffmann-La Roche also uses SKOS annotations like skos:definition or skos:prefLabel to annotate schema elements.

SPARQL PROTOCOL AND RDF QUERY LANGUAGE (SPARQL)

The W3C defines a language, called SPARQL (a recursive acronym), to access and query information stored in RDF. Usually access is provided through a SPARQL endpoint, a network service point that will accept RDF queries. SPARQL defines four types of queries. SELECT queries extract information from RDF data in table format, CONSTRUCT queries extract information from RDF data in RDF format, ASK queries return a boolean result, and DESCRIBE queries return a standard description for a given resource. The keywords of a SPARQL query resemble those of relational queries (SELECT, WHERE, ORDER BY etc.), but the actual data selection is based on graph matching criteria expressed as triple patterns. The following example query selects all Diabetes Rave forms from the Hoffmann-La Roche data collection standards.

Query Editor | Query Library

```
SELECT ?raveForm ?formOrdinal ?formLabel
WHERE {
   ?raveForm rdf:type mms:DataCollectionForm .
   ?raveForm mdr:context mms:Model.PD-Biometrics.Rave .
   ?raveForm mms:extendsModelComponent/mms:subDataElementDomainOf daco:DED.Diabetes .
   ?raveForm mms:extendsModelComponent/mms:domainOrdinal ?formOrdinal .
   ?raveForm mms:extendsModelComponent/mms:domainLabel ?formLabel .
}
ORDER BY ?formOrdinal
```

Figure 5. Example SPARQL Query

The WHERE clause is a graph matching pattern specified as a list of triples containing variables such as ?raveForm. The query engine will search for all sub-graphs that match the pattern and return the results for the corresponding matched variables. Stepping through this example:

1. The query finds any resource of the type Data Collection Form in the context Biometrics Rave Model that is within the Diabetes domain and stores that resource as a variable ?raveform.
2. For each matched resource, additional matches for form ordinal and form label properties searched for and returned as variables ?formOrdinal and ?formLabel respectively.

Note that ?raveForm is a variable used to describe the pattern, i.e. it is only a placeholder name and has no further meaning. Calling it ?raveForm is just very convenient to make the query more human readable.

The query yields the following result:

raveForm	formOrdinal	formLabel
Ext.Form.Diabetes.History	50201	Diabetes History
Ext.Form.Diabetes.HypoglycemicEvents	50202	Hypoglycemic Events
Ext.Form.Diabetes.HomeGlucometerReadings	50203	Home Glucometer Readings (for hypoglycemic events)
Ext.Form.Diabetes.AllergicReactionEvents	50204	Allergic Reaction Events
Ext.Form.Diabetes.PancreatitisEvents	50205	Pancreatitis Events

Figure 6. Example SPARQL Result Set

Possibly the greatest asset of SPARQL 1.1 queries is that the same query can access multiple RDF graphs in multiple RDF data sets that are managed by different service endpoints. The combination of linked data principles and federated query enables data integration on a level that is otherwise extremely difficult to achieve.

ISO 11179 FOR METADATA REGISTRIES (MDR)

The W3C semantic standards provide powerful means to define, describe, link, and access data. The next step is to apply these standards to the representation of CDISC-based clinical data standards. Please note, this is not the representation of the clinical trial data itself, but information about its structure and usage, called metadata.

Metadata is information about how clinical trial data is modeled, e.g. descriptions of data collection (CDASH) domains and variables, data tabulation (SDTM) domains and variables and analysis (ADaM) domains and variables. As such, the next step is to create a model that can handle this metadata, i.e. a model to manage models. We call this a Metamodel.

Developing a Metamodel does not require reinvention of the proverbial wheel. Two sources are quite useful. The Object Management Group (OMG) has created a reference framework to talk about metadata. The OMG distinguishes four levels.

Level 3	Metamodel Specification
Level 2	Metamodel
Level 1	Model
Level 0	Data

The Metamodel needs to handle concepts like data element domains, data elements, value lists, and value list elements. Additionally, the Metamodel must enable the managing and curating of information in a systematic way. This includes registering and versioning metadata elements.

ISO 11179 is an extensive standard for Metadata Registries in six parts that addresses this problem domain. The following will not describe all parts in all details, but focus on those issues of immediate relevance to designing and implementing an MDR. The key parts are the Registry Metamodel (Part 3), which deals with the Metamodel itself, and Registration (Part 6), which deals with issues around registering, versioning, and curation.

ISO 11179 PART 3 REGISTRY METAMODEL

The Metamodel splits into two layers. The core of the Registry Metamodel is the distinction between the operational layer and the conceptual layer.

The operational layer (Data Element, Value Domain, Permissible Value) describes the representation of data elements and their code lists. This can be used to describe data standards such as CDASH, SDTM, and ADaM. A **Data Element** is generically defined as a unit of data uniquely identified within a **Context**, which may be a file, data set, data model, data collection form or anything else that may provide a context for data elements. A **Value Domain** can be enumerated, in which case it is a set of **Permissible Value** instances defined at the operational layer.

The second layer is the conceptual layer. Roughly speaking, each data element and value domain gets linked to a corresponding concept in the conceptual layer. ISO 11179 leaves very much open how to further deal with concepts. A basic decomposition is to split a data element concept (e.g. Adverse Event Outcome) into an object class (Adverse Event) and a property (Outcome). The conceptual layer could be restricted to a simple model in the style of SKOS or go as far as creating a comprehensive ontology. In short, the conceptual layer provides the corresponding conceptual meaning of the metadata elements defined in the operational layer.

The Hoffmann-La Roche Metamodel implementation and the FDA/PhUSE Semantic Technology project have created an OWL model to capture the key features of ISO 11179 Part 3. The FDA/PhUSE models have been slightly improved to better deal with context relationships. The resulting OWL class hierarchy is shown below.

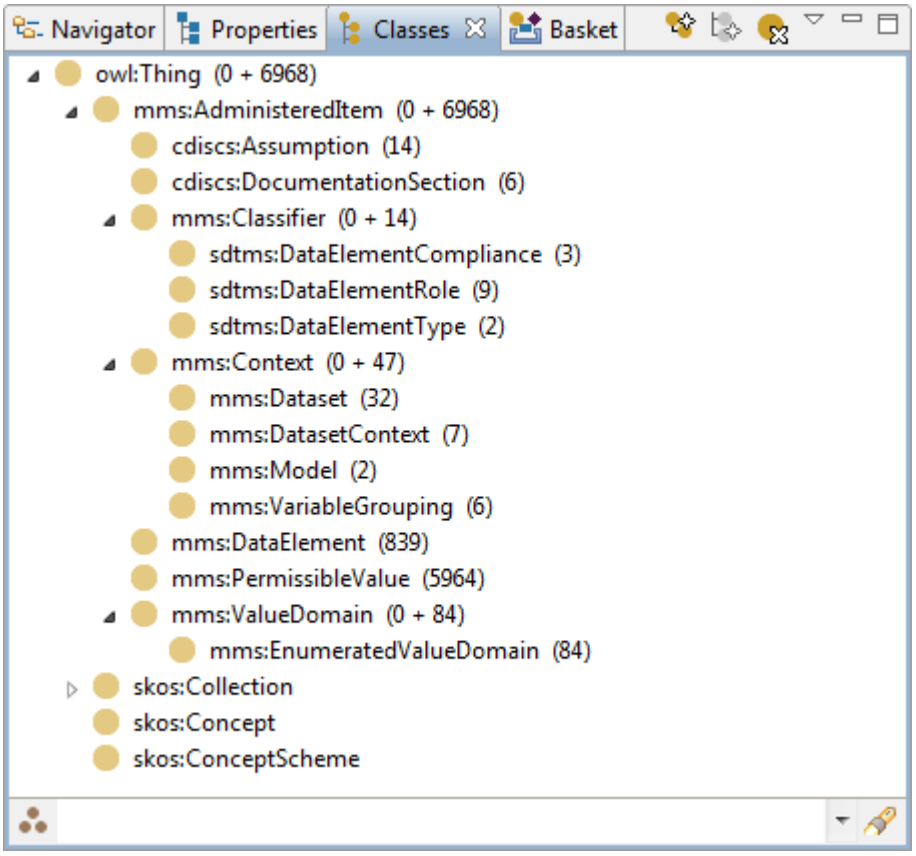

Figure 7. ISO 11179 OWL Class Diagram

This particular model has been populated with a description of the SDTMIG 3.1.2 model. The numbers indicate the number of instances of each class, e.g. there are two instances of the Model class (SDTM 1.2 and SDTMIG 3.1.2) and overall 839 data elements across both models, 84 code lists and 5964 code list elements. The following is a form-based resource description of the SDTMIG 3.1.2. Model resource.

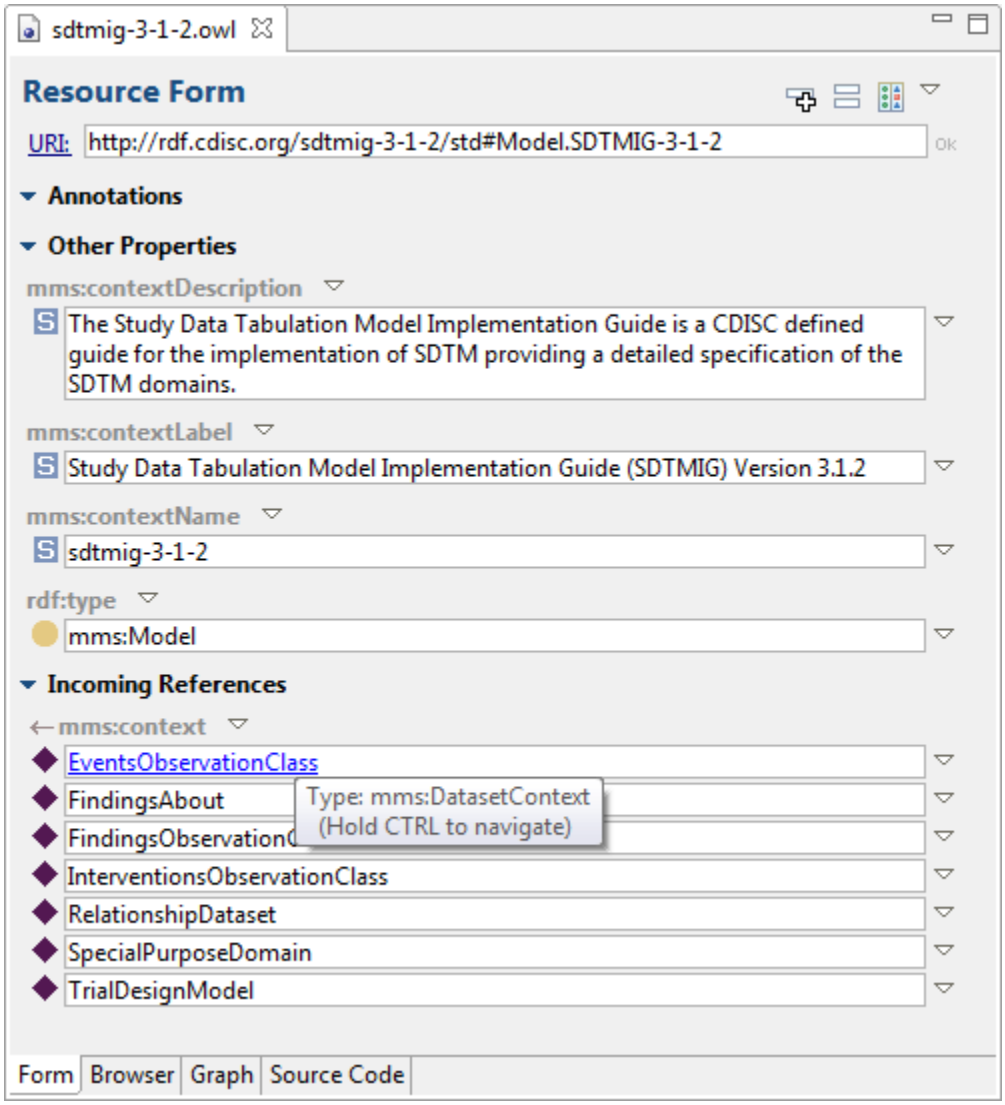

Figure 8. SDTMIG 3.1.2

This is a form representation of the underlying RDF graph. The URI at the top is the subject of a triple for each property on the form. The URI is the object of a triple for each of the incoming references. Navigating the linked RDF resources is a straightforward exercise. For example, navigate to the Events Observation Class resource, further (not shown) to the AE domain resource, and further to the AEOUT domain variable resource.

ISO 11179 PART 6 REGISTRATION

Registration deals with administering MDR items in terms of governance, versioning, and lifecycle management.

The key element in this part of the model is that each MDR item is an **Administered Item** with an **Administration Record**. If an item needs to be versioned, a new Administered Item is created with a link to its prior version, with each version having its own valid time window defined by the interval [effectiveDate, untilDate]. The lifecycle registration status provides several levels as an item moves from Proposed to Candidate to Standard. Items are never removed, but can be Retired or Superseded by another Administered Item. Versioning of an Administered Item with relationships requires some additional consideration; however, this is a solvable issue.

The following diagram is part of the Hoffmann-La Roche semantic model that represents ISO 11179 Part 6.

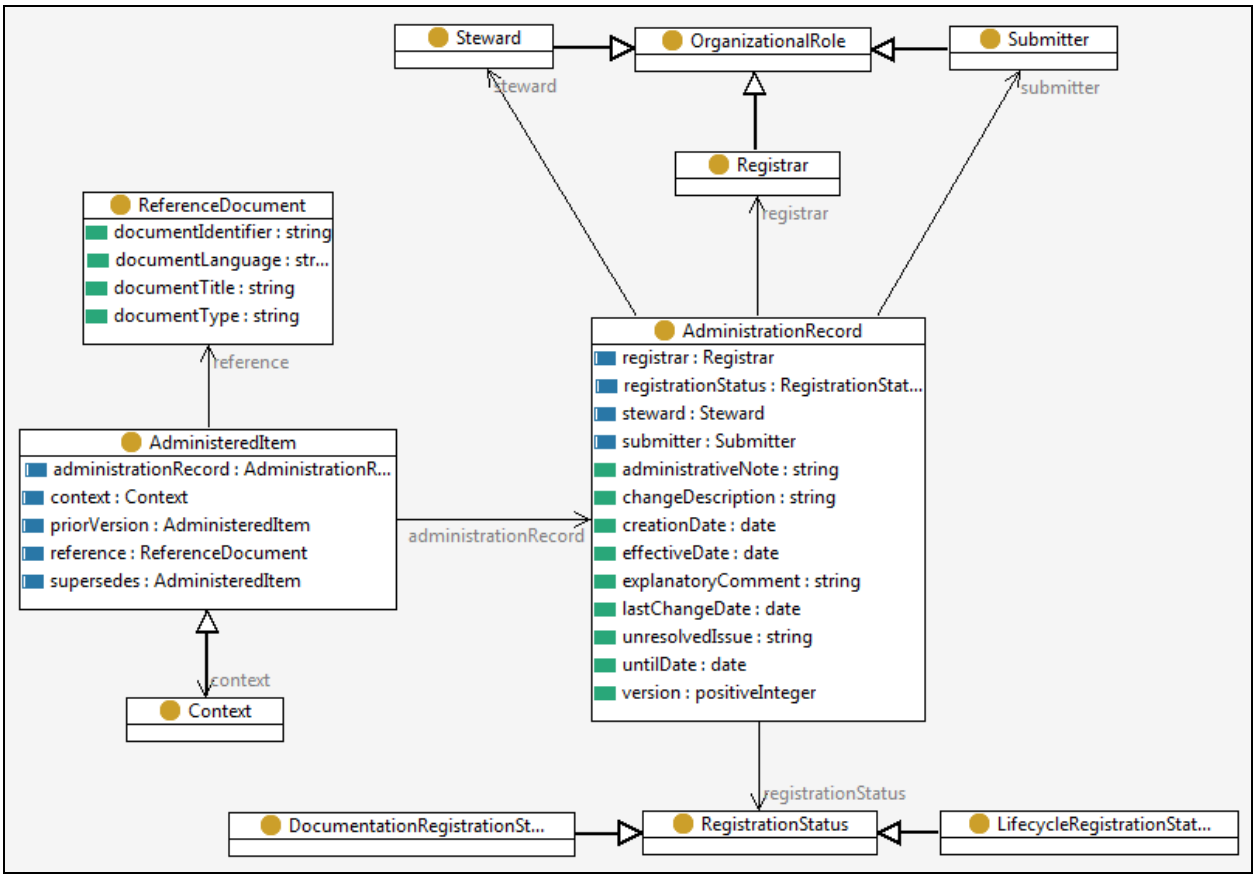

Figure 9. OWL Class Diagram of ISO 11179 Part 6

With all the design trade-offs one can make, registration should be considered the most critical piece of any MDR. When it comes to the registration process, it is important not only to define the workflow processes, but also to make the lifecycle management an integrated part of the model.

THE HOFFMANN-LA ROCHE DATA STANDARDS IMPLEMENTATION

As indicated previously, the Hoffmann-La Roche clinical trial data standards are based on CDISC and built on an ISO 11179 foundation represented in RDF, resulting in the following Information Model.

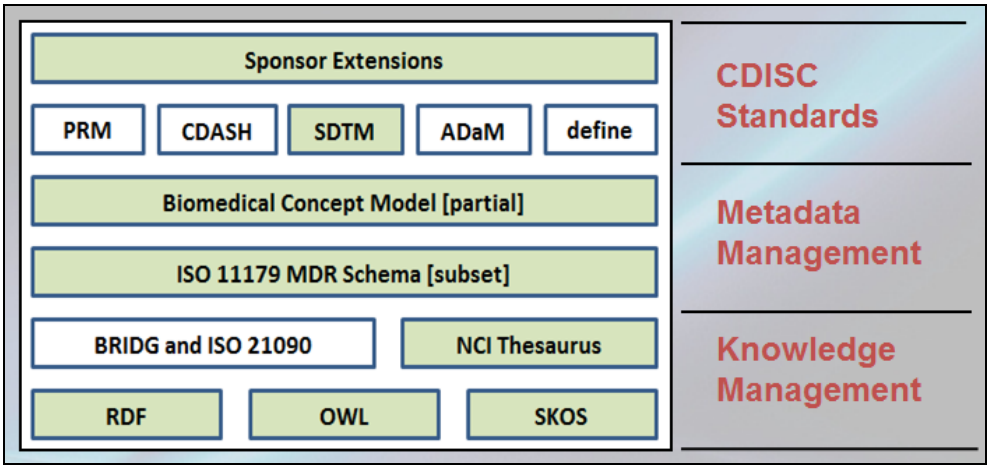

Figure 10. Hoffmann-La Roche Information Model for Clinical Data Standards

Please note, CDASH has not been fully implemented due to data management system integrations; however, sponsor-defined data collection standards across all therapeutic areas of sponsor interest have been defined. The scope of data standards implementation within Hoffmann-La Roche is currently expanding to include protocol and analysis standards.

MDR IMPLEMENTATION

In terms of system components, this information model is deployed in a Metadata Registry called the Global Data Standards Repository (GDSR) as indicated below.

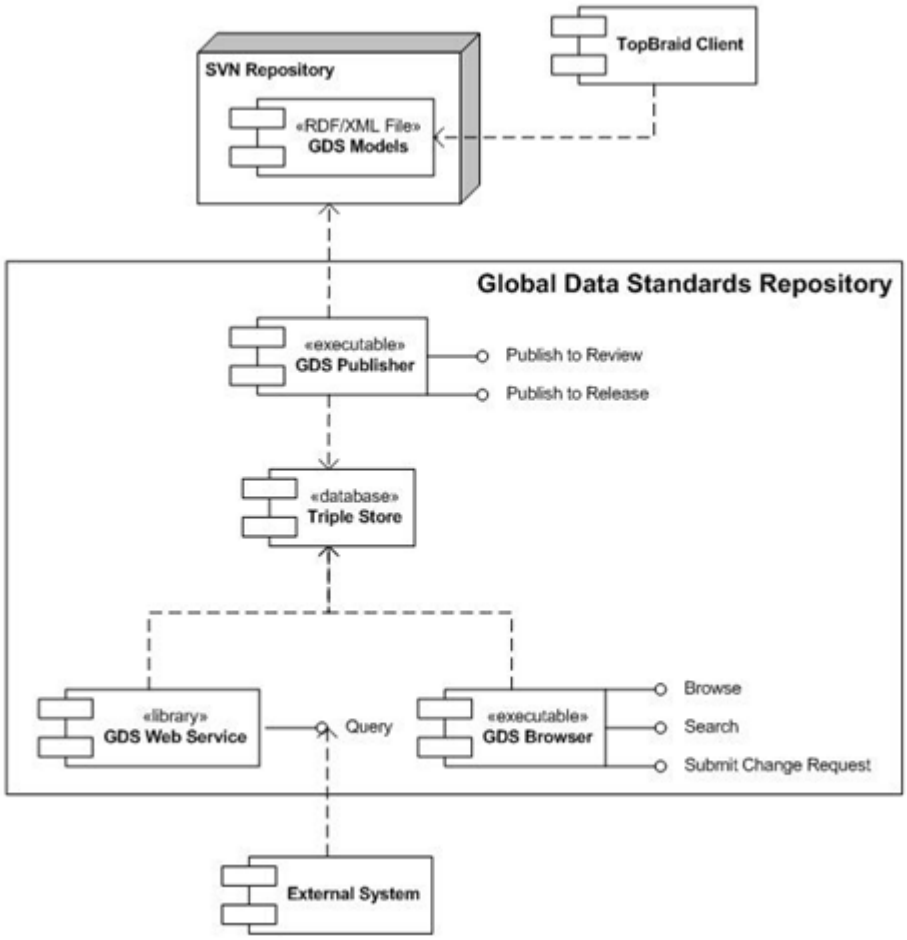

Figure 11. Hoffmann-La Roche MDR Components

The schemas and data standards content are managed by an Information Architect group within the Data Standards Office using a semantic modeling workbench. New versions can be uploaded into a so-called triple store that manages RDF data sets. Once reviewed, the data standards are published into a release area where end users can access the data.

Information can be accessed through a web-based browser that includes search functionality, or via SOAP based web services. The web services allow client programs to extract data standards and use that information to automate certain tasks. For example, data collection standards can be extracted and transformed into a format defined by Medidata for direct upload into Medidata Rave and instantiation of a Global Volume.

The system provides full support for item-level versioning, i.e. data standards can be updated and published on a continuous basis, and data standards content can be accessed based on any versioning date. This applies to all access methods (browse, search, web services) and guarantees full backward compatibility for users of the data standards.

RDF MODEL DRIVEN REST SERVICES AND SEARCH

The RDF models and initial data standards content were created in 2010 by the Hoffmann-La Roche Data Standards Office, the first release of the MDR implementation described in the previous section occurred in 2011, and since April 2013 it is in its third release.

One shortcoming was the conflicting situation of full flexibility of the modeling and content—it is easy to change, add, and link things with RDF—and fairly hard-coded ways to access the information through browsing, search, and web services. As such, a change in the schema is not automatically visible in the end-user applications and requires changes in the validated, deployed code base. Quickly reacting to change has proved very challenging in a validated environment.

These difficulties have led to considerations to not only store the schemas and the data in RDF, but also to model and store in RDF the specifications of how to access the data, for browsing (UI), search, and web service access. GDSR Release 4, now close to UAT and scheduled to be released at the end of September, will be capable of providing those RDF model driven features for search and REST based web services.

The key to achieve this is a small OWL ontology that describes facets, i.e. views of resources based on certain criteria such as class membership. The GDSR runtime system can read these facets and use these descriptions to dynamically return the correct information for any request to get a resource named by a URI. The example below is the partial response to a request to acquire the description of the Adverse Event Outcome data collection field.

```
<raveField xmlns="http://gdsr.roche.com/ws/sim/facet.dc.raveField.default"
           uri="http://gdsr.roche.com/pd-biometrics/rave#Ext.Field.AE.Outcome">
    <ordinal>15</ordinal>
    <crfLabel>Outcome of AE</crfLabel>
    <dataType>xsd:string</dataType>
    <fieldPolicy>Study Build Mandatory Field</fieldPolicy>
    <formControl>Dropdown Box</formControl>
    <logFormHeaderText>AE OUTCOME</logFormHeaderText>
    <preferredLabel>Adverse Event Outcome</preferredLabel>
    <studyBuildNote>SDTM codelist: C66768</studyBuildNote>
    <hiddenField>false</hiddenField>
    <protectedField>false</protectedField>
    <repeatingField>false</repeatingField>
    <raveFormat>$32</raveFormat>
    <raveFieldOid>AEOUT</raveFieldOid>
    <raveSasLabel>AE Outcome</raveSasLabel>
    <raveDictionaryName>AE_OUTCOME_V1</raveDictionaryName>
    <raveIntegrationField>false</raveIntegrationField>
    <helpTextList>
        <helpText uri="http://gdsr.roche.com/pd-biometrics/data-collection#Help.AE.10">
        <ordinal>1</ordinal>
        <annotationText>Select the outcome of each adverse event at the time of final assessment.</annotationText>
        </helpText>
    </helpTextList>
    <sdtmAnnotationList>
        <sdtmAnnotation uri="http://gdsr.roche.com/pd-biometrics/data-collection#Annotation.SDTM-3-1-2.AE.15">
        <ordinal>1</ordinal>
        <annotationText>AE.AEOUT</annotationText>
        </sdtmAnnotation>
    </sdtmAnnotationList>
    <dictionaryEntryList>
        <dictionaryEntry uri="http://gdsr.roche.com/pd-biometrics/data-collection#DVC.AdverseEvent.Outcome.001">
            <ordinal>1</ordinal>
            <codedData>FATAL</codedData>
            <userDataString>Fatal</userDataString>
            <studyBuildPolicy>Study Build Mandatory Value</studyBuildPolicy>
        </dictionaryEntry>
```

Figure 12. XML Response of a Get URI Request

In this example, a request for a resource with URI http://gdsr.roche.com/pd-biometrics/rave#Ext.Field.AE.Outcome was issued. The runtime system picked up a default facet based on class membership of this resource, and using the facet information, the complete response was determined and built at runtime. The facet completely determines the response including elements, attributes, their order, and the RDF property paths to fetch the data. Facets themselves are represented by RDF resources and are therefore completely configurable. Facets can also be composed to describe contained resources, e.g. the Adverse Event Outcome field contains help text, an SDTM annotation, and dictionary entries. These entities are also represented as RDF resources, each with a URI, and have their own facet that describes their representation in this response.

Users can request resource descriptions to be returned in XML, JSON, CSV, and resource schema information in XSD and REST-ready Java classes. Additionally, an XSLT or XSL-FO transformation can be associated with a facet. In that case, the XML result will be processed by this transformation before being returned to the user. Finally, REST requests for resources can also provide a request parameter for the date at which the resource was valid. Based on the date and the ISO 11179 administration record, the correct version of the resource will be returned.

Facets also enable model driven search. Simply declaring a facet as searchable surfaces all resources described by that facet in a search index such as a Lucene index for application specific search and Google Search Appliance for across system search.

Model driven UI has only partially been achieved. It is now possible to partly configure the UI using external configuration files, but they are still limited in scope and do not cover the full spectrum of UI components. A full resource oriented model driven UI that executes on the client browser using HTML 5, CSS 3, and JavaScript is planned to be implemented in the next release.

WORKFLOW AUTOMATION

Model driven web services will enable Hoffmann-La Roche to further realize the benefits of data standardization and facilitate the development of MDR-based metadata driven workflows. Hoffmann-La Roche has initiated a series of ten experiments designed to explicitly document the benefits of data standards and metadata driven workflows. These experiments will quantify time and resource savings as well as increased accuracy relative to current manual processes.

Example experiments include the following:

- Automated creation of an operational CRF view of data collection standards in PDF format.
- Automated creation of a CRF view of data collection standards based on the regulatory submission requirements.
- Automated creation of data collection import files as defined by Medidata for direct upload into Medidata Rave and instantiation of a Global Volume.
- SDTM transformation specification and code generator.
- Representation of a protocol schedule of assessments.
- Automated creation of study-specific documentation including specifications for a visit/form grid and vendor transfer specifications.

OTHER INITIATIVES

Much work on applying semantic technologies in the area of clinical data standards has been done at Hoffmann-La Roche over the past few years. At the same time, industry awareness around the value of metadata and the use of an MDR has clearly taken center stage as can be seen from many conference papers, initiatives, and tentative vendor offerings. The following initiatives provide an opportunity for semantic technologies to play a role in this area.

FDA/PHUSE SEMANTIC TECHNOLOGY PROJECT

The 2013 FDA/PhUSE Computational Science Symposium launched a new working group to explore emerging technologies. The Semantic Technology project is one of four projects in this new Emerging Technologies working group. The project has garnered significant interest and provides a platform for those in the life sciences space to share their experiences. Additionally, sub-projects have been launched to apply semantic standards to different opportunities and challenges in life sciences. The first sub-project focused on representing existing CDISC foundational standards in RDF. This has been taken up by four teams and draft deliverables were finished by mid-August. These deliverables consist of RDF representations of CDASH 1.1, SDTM 1.2, SDTMIG 3.1.2, SDTM 1.3, SDTIG 3.1.3, SENDIG 3.0, ADaM 2.1, and ADaMIG 1.0. More information can be obtained at the PhUSE Discussion Club.

For the second half of the year we are investigating to have teams do work around the following topics.

- Representing regulations and guidance in RDF
- Representing CDISC conformance checks in RDF
- Representing clinical trial data in RDF and investigating toolsets to access the data in RDF
- Representing the CDISC PRM in RDF

If you are interested in joining any of these teams, please contact Scott Bahlavooni. More information can be found on the PhUSE Wiki.

CDISC CONTROLLED TERMINOLOGY

On the brink between ceasing the precursor CDISC2RDF project and starting the PhUSE Semantic Technology project, the NCI took interest in earlier work around representing CDISC Controlled Terminology in RDF. The NCI has now taken ownership of this and CDISC Controlled Terminology can be downloaded in RDF from the NCI web site by following the link at http://www.cdisc.org/terminology. The schemas are identical to those used by the FDA/PhUSE deliverables to represent existing CDISC standards in RDF.

CDISC BRIDG AND SHARE

Since BRIDG 3.2 there is also an OWL/XML format of BRIDG available. This is pretty much a one to one translation, roughly mapping each BRIDG UML class to an OWL class and each BRIDG UML relation to an RDF predicate.

CDISC has been working on its own implementation of an MDR, called SHARE, to enable more streamlined standards development and to be ready to handle larger volumes related to Therapeutic Area standards development. The SHARE vendor implementation that started this summer is based on ISO 11179 and part of the requirements is to provide a data standards export package in RDF.

CONCLUSION

We hope we could make a convincing case for the fundamental value that RDF can provide for modeling and building systems to manage ISO 11179 and CDISC based clinical data standards. RDF has often been named as an emerging technology, but we think the standards and toolsets are available and robust enough to create real production systems in the corporate world. The Hoffmann-La Roche implementation has proven it can be done. We are also encouraged by the increased attention this technology has attracted over the last few years and we are hopeful that more work in this area can support the goal of achieving true semantic interoperability.

REFERENCES

1. Go to w3.org/2001/sw/wiki/Main_Page for an entry page to the W3C semantic web standards.
2. Go to metadata-standards.org/11179 for the six parts reference documents on ISO 11179.
3. Go to cdisc.org for comprehensive documentation on all CDISC standards.
4. Go to omg.org/spec for a directory of all OMG specifications.
5. Go to cabig.nci.nih.gov/concepts/EVS/ to see what the National Cancer Institute (NCI) is doing in the area of controlled terminologies and ontology modeling.
6. Go to bioontology.org to visit the National Center for Biomedical Ontology (NCBO), a great resource for biomedical ontologies and related technologies. It provides a repository and federated search across a large number of biomedical ontologies.

ACKNOWLEDGMENTS

We would like to acknowledge all the hard work and continued dedication by the people in the Hoffmann-La Roche Data Standards Office (DSO). In this paper we have mainly talked about technical aspects, but it takes much more to create successful data standards within a large pharma corporation, not least a huge amount of standards knowledge, subject matter expertise, and dedicated persistence to make standards governance work.

We also acknowledge the great work by the FDA/PhUSE Semantic Technology project volunteers. They have delivered RDF representations of a large number of existing CDISC standards within a very short time span. In particular, we like to thank Mitra Rocca, Phil Ashworth, Geoff Low, and Josephine Gough for their team lead efforts and the many opportunities for engaging conversations.

We are grateful to Josephine Gough for reviewing this paper and providing suggestions.

RECOMMENDED READING

1. Dean Allemang and Jim Hendler. Semantic Web for the Working Ontologist. Second Edition. Morgan Kaufmann, 2011. This is an excellent book, well-written, specifically on the modeling aspects of RDF and OWL in the sense that we indicated in this paper. If you are going to read one book on the subject, we suggest to make it this one.

2. Christopher Walton. Agency and the Semantic Web. Oxford University Press, 2007. This book gives a broad outlook on knowledge systems and the semantic web, including more academic background on the computational aspects of the subject.

3. Dragan Gasevic, Dragan Djuric, and Vladan Devedzic. Model Driven Engineering and Ontology Management. Second Edition. Springer, 2009. This book provides valuable insight on knowledge engineering and the relationship between the different modeling spaces.