PhUSE US Connect 2018

Paper TT17

Transforming Clinical Trials with Linked Data

Armando Oliva, Semantica LLC, Fort Lauderdale, USA

Tim Williams, UCB Biosciences Inc., Raleigh, USA

Abstract

The pharmaceutical industry continues to be plagued by data integration and management challenges across the clinical trial data life cycle. Considerable progress has been made in recent years with the implementation of CDISC standards. Historically, standards focused on distinct segments of the clinical trial process: study design, submission, publication. To provide a future-proof solution, these standards must be adapted and integrated holistically and consistently across all use cases.

Linked Data provides a potential solution by representing clinical trial concepts at their atomic level, then leveraging ontological classification and rules integration. This paper reports results from the PhUSE project "Clinical Trials Data as RDF." SDTM data was converted to Linked Data based on CDISC and custom ontologies, then reassembled into high-quality, submission-ready data sets. The approach has several advantages, including the inextricable representation of data and their meaning in ways not possible in traditional approaches.

Introduction

For more than 15 years, we have witnessed the gradual, and more recently, rapid adoption and implementation of CDISC[1] standards. This has in general been a great success story. For example, the implementation of the Study Data Tabulation Model (SDTM) for regulatory submission datasets to the U.S. Food and Drug Administration (FDA) has led to a new generation of automated tools to process and analyze the data resulting in improvements and efficiencies in scientific and regulatory review. During the “Dark Ages” before data standards, the industry was barely crawling with respect to automated data management and analysis processes, but we can universally acknowledge that the industry is now walking, and at a fairly brisk pace! However, remaining problems prevent the industry from running. Our industry continues to face data integration and management challenges despite the availability of data standards. Inconsistencies in standards implementation is only one of several reasons that we fail to achieve an optimal level of computable semantic interoperability (CSO). This project explores the use of Linked Data to resolve many of these issues.

Linked Data is defined as a method of publishing structured data so that it can be interlinked and become more useful through semantic queries.[2] Linked Data provides a potential solution to today’s problems by representing clinical trial concepts at their atomic level, expressing more semantics explicitly, then leveraging ontological classification and rules integration. The Resource Description Framework (RDF), a World Wide Web Consortium (W3C) standard, is an established approach to achieve Linked Data solutions. This paper reports results from the PhUSE project "Clinical Trials Data as RDF." Study data was converted to Linked Data based on CDISC and custom ontologies, then reassembled into high-quality, submission-ready SDTM data sets. This approach benefits from the ability to define concepts computationally so that inconsistencies in implementation can be minimized. The result is the automated creation of highly structured, highly consistent SDTM data, thereby essentially removing the high variability in SDTM implementation seen today. The approach has several additional advantages, including the inextricable representation of data and their meaning in ways not possible in traditional approaches.

In the world of clinical trials data management and analysis, Linked Data provides the ability to take the industry to the next level of study data management and analysis.

The Problem

The process to create and submit SDTM datasets is slow, highly manual, and error-prone. The instructions on creating valid SDTM datasets are located in human readable PDF documents. It is not unexpected that variability in implementation is widespread as different human interpretations of the instructions are common. Worse still, the instructions are scattered across multiple sources and organizations and formats, making the standards themselves silos. One must know CDISC models, SDTM terminology, MedDRA[3], WHO Drug Dictionary,[4] and other standards, and must know how to integrate them holistically. Add on top of this the fact that standards are continuously evolving, often at different paces from one another, and the implementation challenges are magnified. As an example, take the SDTM variable RACE. How to use this variable is described in the SDTM Implementation Guide published by CDISC.[5] The permissible values for RACE are found elsewhere: in the SDTM terminology document made available by the National Cancer Institute (NCI).[6] The FDA guidance on the collection of race data in clinical trials is yet in a third document.[7] The ability to link RACE in one document with the permissible values for RACE in another document is exactly what Linked Data is designed to do, so that an information system can easily link the two without having to rely on human memory.

As another example, consider the SDTM reference exposure end date (`RFXENDTC`). This is the last known date of exposure to study medication for a given subject in a trial. It is located in the DM (Demographics) domain. In reality, this is a variable that is derived from individual subject exposure records in the EX (Exposure) domain. Because the derivation is not computable, human error results in values for `RFXENDTC` that are not consistent with the more granular exposure data in EX. RDF provides the ability to define this concept computationally so that its derivation is consistent and automated across studies.

A third example is the representation of SDTM concepts that have varying definitions across submissions. Two examples are the reference start date and also the treatment emergent flag for an adverse event. The sponsor-provided definitions are often included in a separate define.xml document, but sometimes the details are buried in the protocol or study report, unavailable to automated systems. RDF addresses this problem by providing the ability to link the concept to its computationally valid definition. When pooling data across studies, it is important to understand whether two variables named the same can be pooled. With RDF, the computer can now assist in that determination.

Objectives

The project intended to show how clinical trial data expressed in RDF can be used to automate the creation of highly standardized SDTM domains. The domains chosen for the pilot were Demographics (DM), supplemental Demographics (SUPPDM), Vital Signs (VS) and the corresponding define.xml file. As a corollary, we intended to demonstrate how both the data and the instructions for SDTM dataset creation can be described in RDF, supporting automated dataset creation and thereby eliminating inconsistent implementations of the standards while substantially speeding up the activity.

To demonstrate the objectives, we needed study data in RDF but found none. Fortunately, the PhUSE Test Dataset Factory Project[8] recently converted data from a study in the public domain to SDTM 3.2. Source datasets are downloadable as SAS transport files from GitHub[9]. Data from the first three patients in this data source was manually converted into RDF to inform study mini-ontology development. The mini-study ontology was then used to craft the process for converting all subject data into RDF (see **Data Conversion**). Starting with SDTM data for the pilot allowed a side by side comparison of the outputs from the manually created source datasets with the automated ontology-based datasets to verify the integrity of the process, which we call “round-tripping.” The intent of the pilot is to eventually eliminate the highly manual SDTM dataset creation process and replace it with a more precise and automated ontology-based creation process, thereby reducing variability, greatly speeding up SDTM dataset creation, and improving standards conformance.

Mini-Ontology Development

Early in the process, we needed a mini-ontology of a clinical trial expressed in OWL (the Web Ontology Language). The word “mini” serves to remind us that, for our work, the ontology represents only a small portion of the data generated in a clinical trial, namely demographic and vital signs data.[10] Initially we considered using BRIDG[11] as the ontology. We had access to BRIDG 4.2 (OWL file) as well as BRIDG 5.0 (no OWL file). Both versions had missing key classes and insufficient representation of relationships between existing classes. For example, there are no classes in BRIDG 5.0 for a Medical Condition or an Assessment (a.k.a. Adjudication). We also found certain modeling constructs as potentially problematic. For example, adverse events are modeled in BRIDG as observations whereas we believe they are Medical Conditions whose identification (e.g. diagnosis) and characterization (e.g. severity) are outcomes of an Assessment of one or more clinical observations. Because of these differences, we created a mini-ontology that uses BRIDG concepts wherever possible but deviates from BRIDG in certain aspects where we felt was necessary.

The mini-ontology is based on several design principles:

1. A clinical trial is fundamentally a collection of activities, each of which has an outcome (e.g. result of a test), and also has a start rule describing when the activity is (or in this case was to be) performed.
2. Represent clinical data according to how the data are generated and used, including the clinical experimental setting: first one observes, then one assesses, then one treats based on a plan.[12] Data for each of these subprocesses must have an unambiguous representation in the graph.
3. Model the data, not the standard, by using classes and property names that subject matter experts can understand and are organized using relationships that make clinical sense.
4. Assume a standards-agnostic approach, allowing data export in various formats including SDTM (the focus of the pilot) and in the future ADaM,[13] FHIR,[14] and others.
5. Build only what is needed for the pilot to create highly standards conformant demographics and vital sign data and related domains including define.xml
6. Add more content to the ontology later. The nature of the graph facilitates incremental development.
7. Focus on flexibility, extensibility, reusability.

The core of the ontology is based on a `:HumanStudySubject`[15] who `:participatesIn` a `:Study`. The `:HumanStudySubject` may be afflicted by one or more `:MedicalCondition`, one of which is the target of an `:InvestigationalDrug`, in which case the `:MedicalCondition` is also an `:Indication`. A `:MedicalCondition` that emerges or worsens following a drug administration is considered an `:AdverseEvent`. A more complete view of the graph is shown in **Figure 1**. Notice the links to external sources (CDISC, MedDRA, etc.), which is one of the distinct advantages of linked data.

The mini-ontology facilitates understanding of the instance data, searching and retrieving using SPARQL,[16] and creating SDTM datasets. For example, the pilot study has one individual resource for each person who participates in the trial. Each person resource is attached to the `:Person` class in the ontology using the `rdf:type` relationship (not shown here).

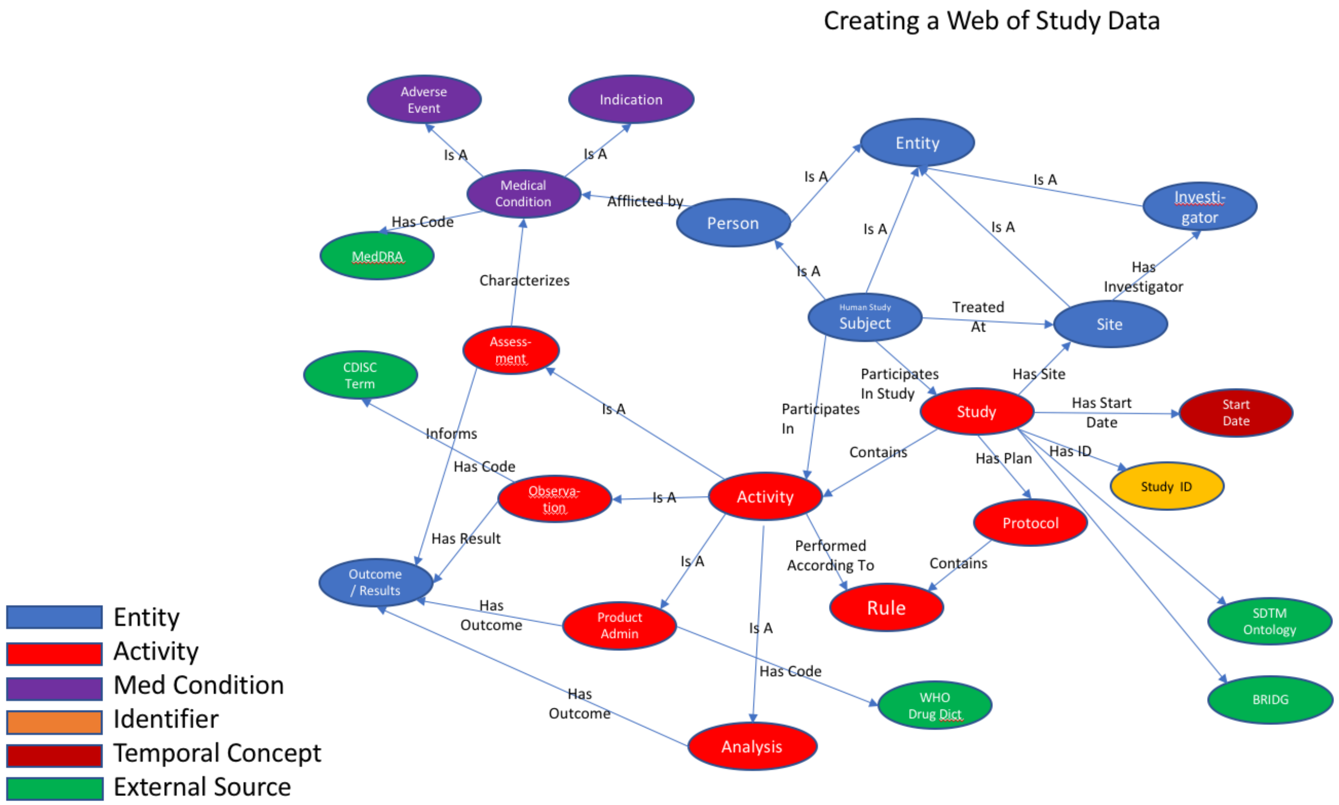


Figure 1: Study "Mini-Ontology"

The mini-ontology can represent and/or derive all SDTM variables found in the DM, VS, and SUPPDM domains. Preliminary experience indicates that the ontology can continue to be modified iteratively to support data from other SDTM domains while maintaining backwards compatibility with the current version.

Data Conversion

The data conversion process from SDTM to RDF was successful in converting data from DM, VS, SUPPDM. The conversion was manual for the first three Subjects in the trial. In the case of VS, there were too many vital signs observations recorded for the first three subjects, i.e. much too many to have been converted manually within a reasonable amount of time. Therefore, only vital sign measurements recorded for the first subject during the first screening visit were converted manually. This turned out to be adequate to test the mini-ontology and support automated data conversion for the remaining subjects.

The automated data conversion process for the remaining subject data and accompanying scripts to transform SDTM to RDF are not considered project deliverables. They are the by-product of translating the source data necessary to develop and test the model. Having an authentic SDTM data source facilitated round-trip validation from the source, to the graph, and then back to the original format. The R application was chosen to convert the SAS transport source data to RDF using the `redland`[17] package. These custom R scripts proved hard to maintain and update as the model evolved. A simpler approach was needed.

Following the 2018 PhUSE Computational Science Symposium (CSS), the data conversion process changed to one that is more scalable and easier to maintain. An overview of the process is shown in **Figure 2**. Like the previous methods, R was used to import the source XPT files and perform data manipulation and imputation. Data required to validate the model that was not available in the original source was created using R or entered manually in supplemental CSV files. For example, a value for death date (`DTHDTC`) and death flag (`DTHFL`) was created in R for one patient in the DM domain even though no deaths were reported in the original study. *Investigator* and *investigatorID* information was not in the original data, so it was created in a CSV file since this information is part of most clinical trial data sets.

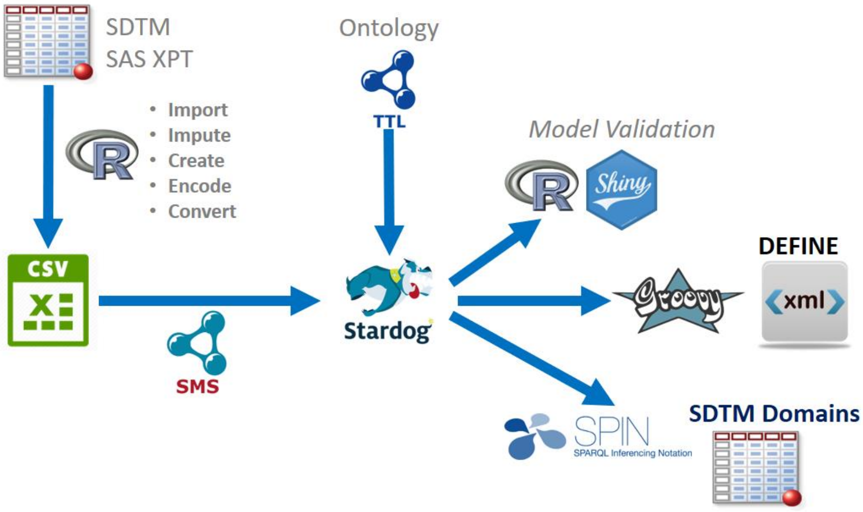
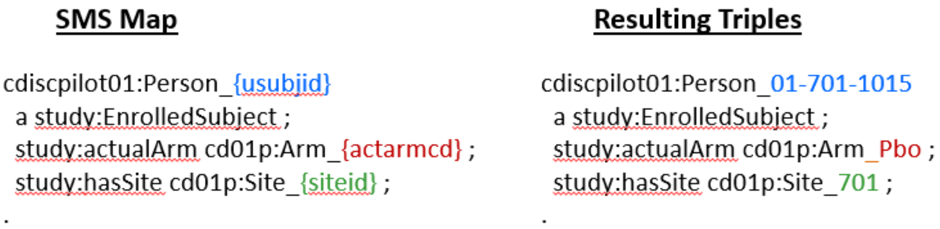


Figure 2: Data Conversion Process

Additional information required in the graph data was not found in the source XPT files. Examples include information like the start rules that occur prior to making an observation or taking a measurement. Consider the case where a blood pressure is taken in the lying position (`VSPOS=SUPINE`, in the VS domain) after the subject assumed a supine position for five minutes. This corresponds to a start rule of `StartRuleLying5`[18] in the graph. These types of rules are part of the study protocol information and can be associated with the results values using OWL 2[19] inferencing. Use of inferencing greatly reduces data redundancy. See the project's GitHub site for more details about this approach.[20]

After the data manipulation is complete, the R data frame for each domain was saved as a comma-separated value (CSV) file that was in turn mapped to the graph database. The W3C standard R2RML "Relational Database to RDF Mapping Language"[21] defines how to map data in a relational or row-by-column format to RDF graphs. Stardog further simplified R2RML as Stardog Mapping Syntax (SMS)[22]. SMS mappings can be converted to R2RML, allowing the project to maintain vendor neutrality for the conversion step. Because SMS is based on Turtle, the structure within map file closely resembles the desired TTL structure for the instance data (see **Figure 3).** When the SMS file is processed, the CSV file is read row-by-row. CSV column names inside the curly braces {} are replaced with the corresponding value from current row. SMS lacks the ability to perform functions like concatenation, so some pre-processing occurred in the R scripts.



SMS Map

```
cdiscpilot01:Person_{usubjid}
 a study:EnrolledSubject ;
 study:actualArm cd01p:Arm_{actarmcd} ;
 study:hasSite cd01p:Site_{siteid} ;
.
```

Resulting Triples

```
cdiscpilot01:Person_01-701-1015
 a study:EnrolledSubject ;
 study:actualArm cd01p:Arm_Pbo ;
 study:hasSite cd01p:Site_701 ;
.
```

Figure 3: SMS Map and Resulting Triples

Other Approaches to Graph Data Conversion

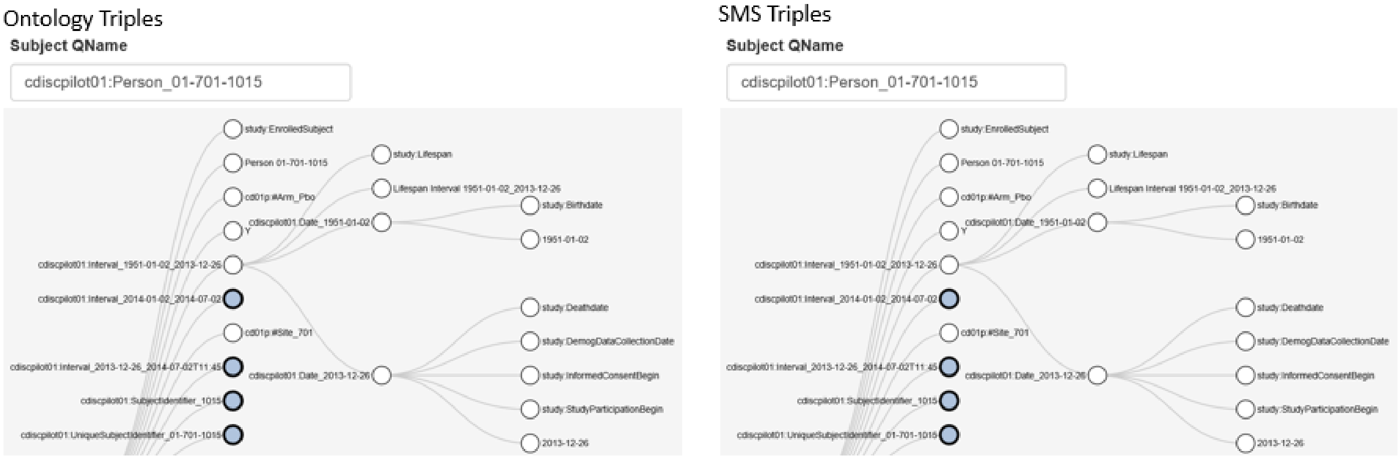
The SAS transport format is an open standard that could, in theory, be mapped directly to the graph database using SMS or R2RML, thus avoiding the intermediate steps for imputation and manipulation. Restructuring of the data could occur within the graph environment. Consider why direct import of XPT files may not be desired. The approach does not match the project philosophy of identifying the atomic entities within the SDTM domains and mapping those entities to the structure defined in the mini-ontology. Converting a row-by-column source to the graph, where the row identifier becomes the *subject*, column identifier the *predicate*, and the cell value the *object* (1) is a fast and efficient way to transform data into a graph but perpetuates many of the problems and limitations inherent in the original structure.[23]

XPT files are commonly created at or near the end of the data lifecycle. In this project the XPT files served as the starting point for data conversion whereas in real-world scenarios the data sources are more likely to be from a relational database system (RDBMS). Data could be supplied directly from the RDBMS to the graph, allowing the original sources to remain in place while the graph becomes the federated data store, bringing together multiple sources with minimal disruption to existing platforms.

Validation of Converted Data

The process of creating the SMS files that match the project ontology was a manual process. SMS TTL files were created by hand using a text editor while cross referencing the ontology, the instance data created from the ontology, and the source CSV files. Not surprisingly, errors often occurred. In the future, it would be very beneficial to have an application that would create the SMS file by visually mapping between ontologies and data sources.

The team used a multi-faceted approach to validate the data created by the SMS mapping, starting with basic SPARQL queries after the data is uploaded to the Stardog triplestore. An R Shiny application graphically compared the ontology triples with the SMS-derived instance data using a collapsible tree rendering of a Stardog `PATHS` query (see **Figure 4**).



Ontology Triples

Subject QName: cdiscpilot01:Person_01-701-1015

SMS Triples

Subject QName: cdiscpilot01:Person_01-701-1015

study:EnrolledSubject
Person 01-701-1015
cd01p:#Arm_Pbo
Y
cdiscpilot01:Date_1951-01-02
cdiscpilot01:Interval_1951-01-02_2013-12-26
cdiscpilot01:Interval_2014-01-02_2014-07-02
cd01p:#Site_701
cdiscpilot01:Interval_2013-12-26_2014-07-02T11:45
cdiscpilot01:Date_2013-12-26
cdiscpilot01:SubjectIdentifier_1015
cdiscpilot01:UniqueSubjectIdentifier_01-701-1015
study:Lifespan
Lifespan Interval 1951-01-02_2013-12-26
study:Birthdate
1951-01-02
study:Deathdate
study:DemogDataCollectionDate
study:InformedConsentBegin
study:StudyParticipationBegin
2013-12-26

Figure 4: Collapsible Tree Rendering of a PATHS Query

In addition to the visual comparison of the tree structure, another R Shiny application compared the values of the triples programmatically to ensure and exact match. Future validation may include the use of Shapes Expressions (ShEx) or Shapes Constraint Language (SHACL).

SDTM Dataset Creation

After we converted the data to RDF, it became possible to query the graph to generate the data for the SDTM datasets. To create the actual datasets, we needed to link to an SDTM ontology. We were able to leverage an SDTM ontology that was already created during a previous PhUSE project, the CDISC to RDF project. We also needed temporal concepts to express dates, times, and intervals. We link to the W3C Time Ontology, which has well defined temporal concepts. This avoids "re-inventing the wheel" with regard to temporal concepts.

For most SDTM variables, there is a 1:1 mapping from the mini-ontology to the dataset. A good example is `BRTHDTC` (the subject's birthdate). The ontology represents birthdate using the approach that every `:Person` has a `:Lifespan` (a temporal interval) and the beginning of that interval is the birth date. The ontology RDF looks like the following. Note the use of the `time:` namespace of the W3C time ontology.

```
:Person        :hasLifeSpan      :Lifespan .
:Lifespan      rdf:type          time:Interval .
time:Interval  time:hasBeginning time:Instant .
time:Interval  time:hasEnd       time:Instant .
```

The SPARQL Query for BRTHDTC is the following:

```
CONSTRUCT {
  ?this sdtm:hasBRTHDTC ?BRTHDTC .
}
WHERE {
  ?this    a                 sdtm:SDTMRecord .
  ?this    sdtm:hasEntity    ?subject .
  ?subject study:hasLifespan ?lfspn .
  ?lfspn   time:hasBeginning ?brthd .
  ?brthd   study:dateTimeInXSDString ?BRTHDTC .
}
```

One can envision a similar query for DTHDTC (death date) and for every other variable where a 1:1 mapping is available. Each query for each variable is represented in the ontology as a spin:rule, thereby allowing the system to automatically retrieve the variable from the knowledgebase using inferencing.

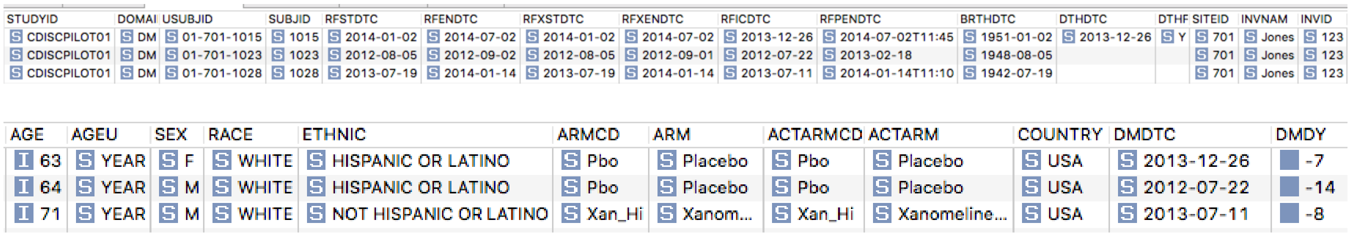
Some variables do not have a 1:1 mapping to the ontology. Often these represent derived variables such as study day. We expressed the standard derivations also in RDF using SPIN. Below is the query for deriving the study day. The query takes the activity date (?date) minus the reference start date (?rfd) to get the duration and adds +1 if the duration is non-negative (since there is no study day 0).

```
SELECT ?studyday
WHERE {
  BIND (xsd:date(SUBSTR(?activityDate, 1, 10)) AS ?date) .
  BIND (xsd:date(SUBSTR(?refDate, 1, 10)) AS ?rfd) .
  BIND (smf:duration("d", ?rfd, ?date) AS ?std) .
  BIND (IF((?std < 0), ?std, (?std + 1)) AS ?studyday) .
}
```

This approach worked well, even for complex derivations like the last date of exposure to investigational drug (RFXSTDC). Although these data are submitted in DM, they are often not reliable because the more granular exposure domain (EX) often records a different date for this variable. The derivation requires examination of each exposure record and selecting the latest date recorded. Once again, by expressing the derivation using SPARQL and embedding it into the mini-ontology using a spin:rule, it enables the system to derive the RFXSTDTC variable consistently and error free. But for the purposes of the pilot (since there is pilot data for this variable) we used the sponsor-provided source data.

Once SPARQL rules extract and/or derive each SDTM variable from the RDF data, another SPARQL query assembles each domain (see Error! Reference source not found.)

Using the methods described previously, we were able to automate the generation of highly standardized SDTM domains (DM, VS, SUPPDM) and the corresponding define.xml content. A comparison with the modified source documents indicated a successful "round-tripping, **Figure 5** below demonstrates the automated output.



STUDYID	DOMAI	USUBJID	SUBJID	RFSTDTC	RFENDTC	RFXSTDTC	RFXENDTC	RFICDTC	RFPENDTC	BRTHDTC	DTHDTC	DTHF	SITEID	INVNAM	INVID
CDISCPILOT01	DM	01-701-1015	1015	2014-01-02	2014-07-02	2014-01-02	2014-07-02	2013-12-26	2014-07-02T11:45	1951-01-02	2013-12-26	Y	701	Jones	123
CDISCPILOT01	DM	01-701-1023	1023	2012-08-05	2012-09-02	2012-08-05	2012-09-01	2012-07-22	2013-02-18	1948-08-05			701	Jones	123
CDISCPILOT01	DM	01-701-1028	1028	2013-07-19	2014-01-14	2013-07-19	2014-01-14	2013-07-11	2014-01-14T11:10	1942-07-19			701	Jones	123

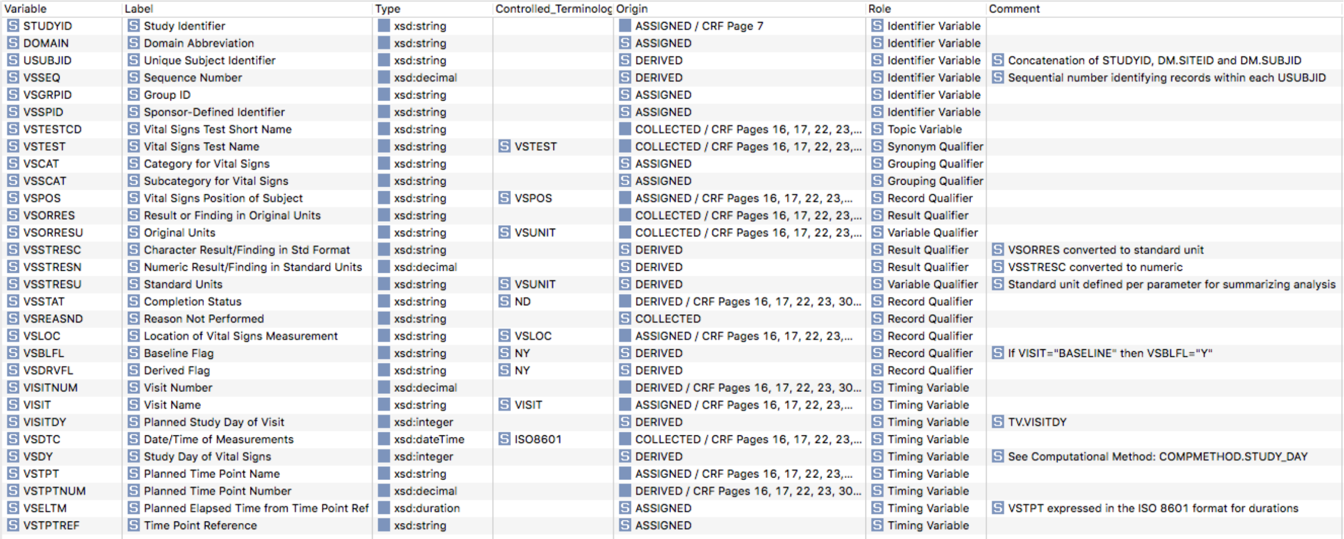
AGE	AGEU	SEX	RACE	ETHNIC	ARMCD	ARM	ACTARMCD	ACTARM	COUNTRY	DMDTC	DMDY
63	YEAR	F	WHITE	HISPANIC OR LATINO	Pbo	Placebo	Pbo	Placebo	USA	2013-12-26	-7
64	YEAR	M	WHITE	HISPANIC OR LATINO	Pbo	Placebo	Pbo	Placebo	USA	2012-07-22	-14
71	YEAR	M	WHITE	NOT HISPANIC OR LATINO	Xan_Hi	Xanom...	Xan_Hi	Xanomeline...	USA	2013-07-11	-8

Figure 5: Automated SDTM dataset creation from source RDF[24]

Define.xml Creation

The define.xml document associated with the pilot data could not be used for conversion to RDF because the document was insufficiently standardized. For example, the Origin column provided either the method for creating the variable value (derived, assigned, **or** the location of the value on the CRF). The two options create semantic ambiguity regarding the definition of Origin. We resolved that by establishing that Origin is either collected, assigned, or derived. If collected, the location on the CRF of the collection is provided.

The first step was to further standardize the metadata for each pilot domain to create a modified define.xml. This served as the source document for the pilot against which effective round-tripping could be demonstrated. Once optimal standardization against the define.xml specification was achieved, we converted the metadata to RDF using the same technique used for the manual conversion process of the study data, as described previously (see **Data Conversion**). SPARQL queries could then be applied to the knowledgebase to automatically generate the define.xml content (see **Figure 6**). As a separate task outside the scope of the pilot is the creation of both define.xml content and format automatically from the knowledgebase.



Variable	Label	Type	Controlled_Terminolog	Origin	Role	Comment
STUDYID	Study Identifier	xsd:string		ASSIGNED / CRF Page 7	Identifier Variable	
DOMAIN	Domain Abbreviation	xsd:string		ASSIGNED	Identifier Variable	
USUBJID	Unique Subject Identifier	xsd:string		DERIVED	Identifier Variable	Concatenation of STUDYID, DM.SITEID and DM.SUBJID
VSSEQ	Sequence Number	xsd:decimal		DERIVED	Identifier Variable	Sequential number identifying records within each USUBJID
VSGRPID	Group ID	xsd:string		ASSIGNED	Identifier Variable	
VSSPID	Sponsor-Defined Identifier	xsd:string		ASSIGNED	Identifier Variable	
VSTESTCD	Vital Signs Test Short Name	xsd:string		COLLECTED / CRF Pages 16, 17, 22, 23,...	Topic Variable	
VSTEST	Vital Signs Test Name	xsd:string	VSTEST	COLLECTED / CRF Pages 16, 17, 22, 23,...	Synonym Qualifier	
VSCAT	Category for Vital Signs	xsd:string		ASSIGNED	Grouping Qualifier	
VSSCAT	Subcategory for Vital Signs	xsd:string		ASSIGNED	Grouping Qualifier	
VSPOS	Vital Signs Position of Subject	xsd:string	VSPOS	ASSIGNED / CRF Pages 16, 17, 22, 23,...	Record Qualifier	
VSORRES	Result or Finding in Original Units	xsd:string		COLLECTED / CRF Pages 16, 17, 22, 23,...	Result Qualifier	
VSORRESU	Original Units	xsd:string	VSUNIT	COLLECTED / CRF Pages 16, 17, 22, 23,...	Variable Qualifier	
VSSTRESC	Character Result/Finding in Std Format	xsd:string		DERIVED	Result Qualifier	VSORRES converted to standard unit
VSSTRESN	Numeric Result/Finding in Standard Units	xsd:decimal		DERIVED	Result Qualifier	VSSTRESC converted to numeric
VSSTRESU	Standard Units	xsd:string	VSUNIT	DERIVED	Variable Qualifier	Standard unit defined per parameter for summarizing analysis
VSSTAT	Completion Status	xsd:string	ND	DERIVED / CRF Pages 16, 17, 22, 23, 30...	Record Qualifier	
VSREASND	Reason Not Performed	xsd:string		COLLECTED	Record Qualifier	
VSLOC	Location of Vital Signs Measurement	xsd:string	VSLOC	ASSIGNED / CRF Pages 16, 17, 22, 23,...	Record Qualifier	
VSBLFL	Baseline Flag	xsd:string	NY	DERIVED	Record Qualifier	If VISIT="BASELINE" then VSBLFL="Y"
VSDRVFL	Derived Flag	xsd:string	NY	DERIVED	Record Qualifier	
VISITNUM	Visit Number	xsd:decimal		DERIVED / CRF Pages 16, 17, 22, 23, 30...	Timing Variable	
VISIT	Visit Name	xsd:string	VISIT	ASSIGNED / CRF Pages 16, 17, 22, 23,...	Timing Variable	
VISITDY	Planned Study Day of Visit	xsd:integer		DERIVED	Timing Variable	TV.VISITDY
VSDTC	Date/Time of Measurements	xsd:dateTime	ISO8601	COLLECTED / CRF Pages 16, 17, 22, 23,...	Timing Variable	
VSDY	Study Day of Vital Signs	xsd:integer		DERIVED	Timing Variable	See Computational Method: COMPMETHOD.STUDY_DAY
VSTPT	Planned Time Point Name	xsd:string		ASSIGNED / CRF Pages 16, 17, 22, 23,...	Timing Variable	
VSTPTNUM	Planned Time Point Number	xsd:decimal		DERIVED / CRF Pages 16, 17, 22, 23, 30...	Timing Variable	
VSELTM	Planned Elapsed Time from Time Point Ref	xsd:duration		ASSIGNED	Timing Variable	VSTPT expressed in the ISO 8601 format for durations
VSTPTREF	Time Point Reference	xsd:string		ASSIGNED	Timing Variable	

Figure 6: Automated define.xml content creation from source RDF

Conclusion

The pilot successfully demonstrated the benefits of using RDF and ontologies as a Linked Data solution for clinical trials. Developing the data conversion process and ontology at the same time strengthened the results from both efforts. Changes in ontology design were tested within a domain prior to wider implementation. Factors related to conversion of real-world instance data were brought forward for consideration in the ontology design. The iterative nature of this approach, while more time-consuming than one focused solely only ontology development, is creating a model that is more representative of the clinical process and its instance data.

Source SDTM data and define.xml metadata from a clinical trial was converted to RDF based on a study mini-ontology, and then back to SDTM and define using automated processes (see
Figure 7). The pilot demonstrated the high degree of automation, and therefore consistency, that one can expect by using RDF as a "back end" solution to replace the current manual, slow, error-prone SDTM creation process.

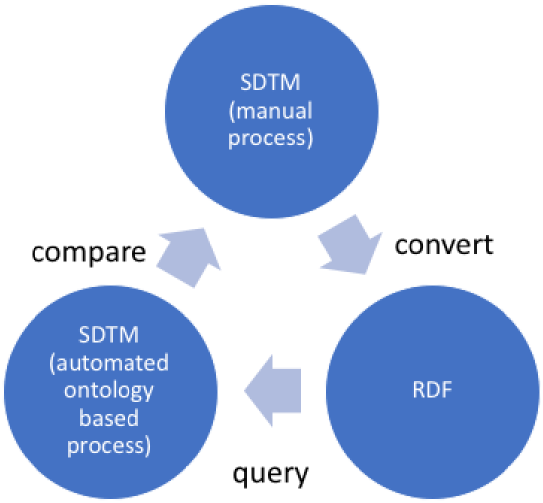


Figure 7: CTD2RDF Round-tripping Process

Benefits of the RDF approach include:

- automated process, literally at the “push of a button”
- faster
- highly standards conformant SDTM data
- decreased variability in standards implementation

Next steps include expanding the mini-ontology so it’s capable of representing and generating all SDTM domains.

The future process to improve the creation of highly standardized SDTM datasets for submission is shown in **Figure 8**. The important change is the use of RDF behind the scenes to automate the process. The desired end result is highly standardized and higher quality of SDTM submissions to the FDA, which means a smoother review process and an optimal and timely regulatory action.

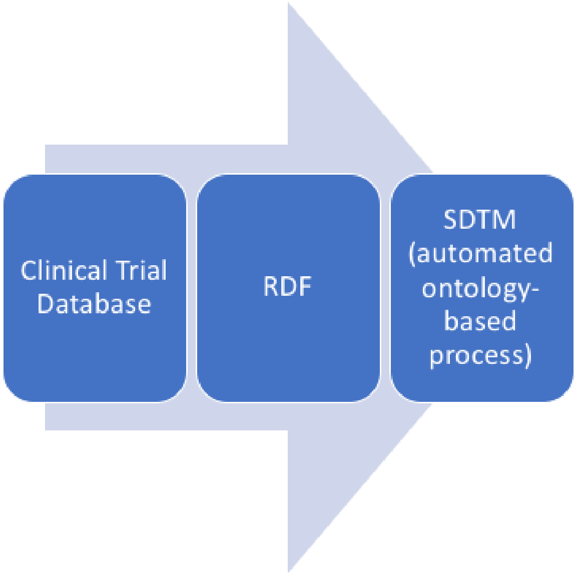


Figure 8: Future State Automated SDTM Dataset Creation

References

1. **Allemang, Dean and Hendler, Jim.** *Semantic Web for the Working Ontologist, 2nd Edition.* 2011.

Acknowledgements

The authors are indebted to the contributions from the "CTDasRDF" PhUSE project team members.

This paper is largely based on free, open-source software and the efforts of volunteers in PhUSE working groups. Please support those who donate their time and expertise through your own collaboration, participation, and promotion of these activities.

Contact Information

Your comments and questions are valued and encouraged. Contact the authors at:

Armando Oliva, M.D.
Semantica LLC
Fort Lauderdale, FL, USA
aoliva@semanticallc.com
@nomini
https://www.linkedin.com/in/aolivamd

Tim Williams
UCB BioSciences, Inc
Raleigh, NC, USA
tim.williams@ucb.com
@NovasTaylor
https://www.linkedin.com/in/timpwilliams

All project files, data, and this paper are available from the project's Github repository: https://github.com/phuse-org/CTDasRDF.
Study instance data: https://raw.githubusercontent.com/phuse-org/ctdasrdf/master/data/rdf/cdiscpilot01.ttl

Brand and product names are trademarks of their respective companies.

[1] Clinical Data Interchange Standards Consortium. See https://www.cdisc.org

[2] See https://en.wikipedia.org/wiki/Linked_data

[3] Medical Dictionary for Regulatory Activities. See http://www.meddra.org

[4] World Health Organization Drug Dictionary. See https://www.who-umc.org/whodrug/whodrug-portfolio/whodrug-global/

[5] See https://www.cdisc.org/standards/foundational/sdtmig

[6] See https://www.cancer.gov/research/resources/terminology/cdisc

[7] See https://www.fda.gov/downloads/RegulatoryInformation/Guidances/ucm126396.pdf

[8] See http://www.phusewiki.org/wiki/index.php?title=WG5_Project_09

[9] See https://github.com/phuse-org/phuse-scripts/tree/master/data/sdtm/updated_cdiscpilot

[10] During this process, we noticed a substantial amount of missing data in the pilot data, such as no birth or death dates, and missing investigator names and IDs. Because we wanted to test the entire ontology, we imputed a small amount of missing data in original source datasets to support a robust testing process. In the case of the define.xml file, the original was insufficiently standardized, so an additional level of standardization was imposed so that the meta-data could consistently be represented in the ontology.

[11] See https://bridgmodel.nci.nih.gov

[12] These sequential subprocesses are taught universally in medical schools by the mnemonic “SOAP” (subjective observations, objective observations, assessments, (treatment) plan, and described here as the “clinical data lifecycle.”

[13] See https://www.cdisc.org/standards/foundational/adam

[14] See https://www.hl7.org/fhir/overview.html

[15] We represent an RDF resource name starting with a colon and use CamelCase for classes and predicates/relationships.

[16] See https://www.w3.org/TR/rdf-sparql-query/

[17] See https://cran.r-project.org/web/packages/redland/index.html

[18] This start rule states that the target activity (BP measurement) takes place only after the subject has been in the supine position for five minutes.

[19] See https://www.w3.org/TR/owl2-overview/

[20] See https://github.com/phuse-org/CTDasRDF/blob/master/DataMappingAndConversion.md

[21] See https://www.w3.org/TR/r2rml/

[22] See http://docs.stardog.com/#_stardog_mapping_syntax

[23] For example, a simple row by column mapping of the DM domain results in AGEU (age unit) being represented in RDF as a property of the Subject. This is non-sensical since in reality AGEU is a property of the subject’s age!

[24] Data shown only for the first 3 subjects