Generating Analysis Results and Metadata – report from a PhUSE CS project

Marc Andersen, StatGroup ApS, Copenhagen, Denmark
Marcelina Hungria, DIcore Group, LLC, NJ, USA
Suhas R. Sanjee, Merck & Co., Inc., Kenilworth, NJ USA

Overview

- Introduction
- Material
- Scope
- Process
  - Generate Analysis Results Dataset
  - Presentation from RDF Data Cube
- Traceability
- Putting it All Together: Application
- Evaluation
- Conclusion

Introduction

- Use of W3C Semantic standards for clinical & non-clinical trial data life cycle
- PhUSE CS Semantic Technology Group
  - Technical specification
  - White paper
  - R Package
- Store analysis results as RDF data cubes
- Potential features and benefits are discussed in this presentation

Material

- 2013 CDISC Pilot Submission Package (http://www.cdisc.org/sdtmadam-pilot-project)
  - ADaM Datasets
  - CSR
  - Define-xml

SCOPE

Tables Reproduced from CDISC Pilot Project

Table	Title	ADAM
14-1.01	Summary of Populations	ADSL
14-1.02	Summary of End of Study Data	ADSL
14-1.03	Summary of Number of Subjects by Site	ADSL
14-2.01	Summary of Disposition	ADSL
14-3.01	Primary Endpoint Analysis: ADAS Cog (11) - Change from Baseline to Week 24 - LOCF	ADQSADAS
14-5.01	Incidence of Treatment Emergent Adverse Events by Treatment Group	ADAE

Process

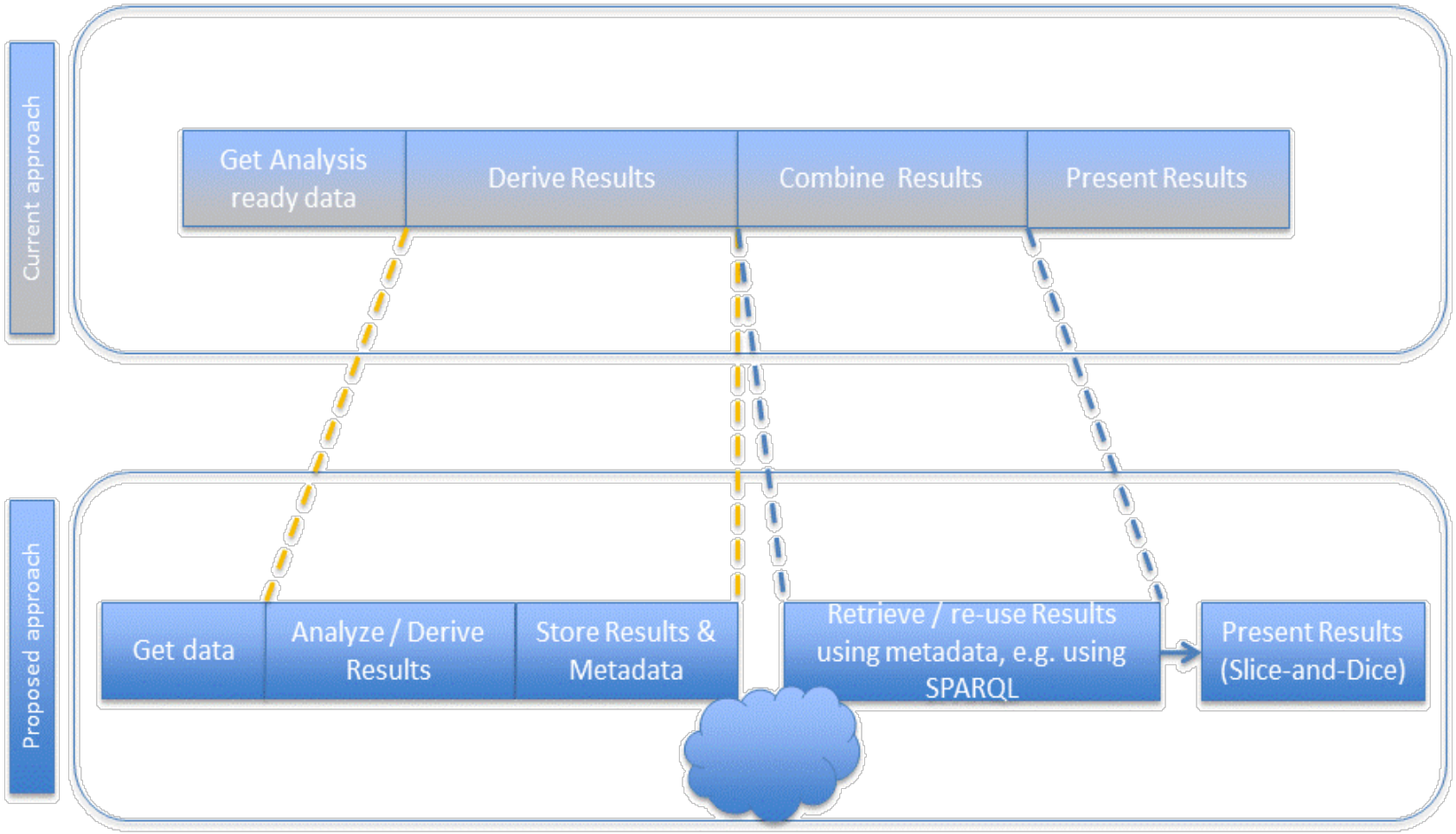
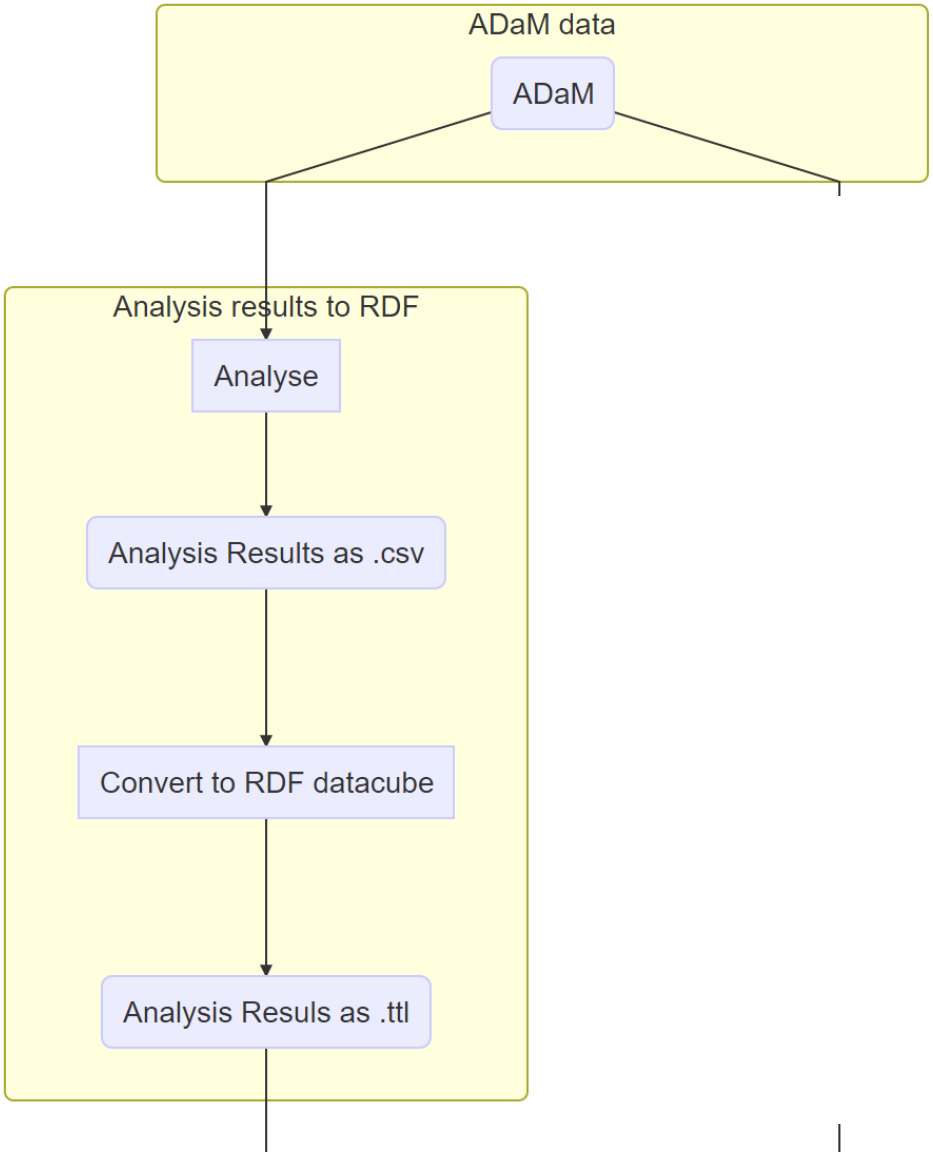


Figure 1: Process Flow of Proposed Approach


ADaM data
ADaM
Analysis results to RDF
Analyse
Analysis Results as .csv
Convert to RDF datacube
Analysis Resuls as .ttl

Generation of Analysis Results Dataset

- Generate summary statistics in SAS using PROC TABULATE (Link)
- Store the analysis results using ODS output and export to a .csv file (snapshot below).

ittfl	saffl	efffl	comp24fl	compfl	trt01p	procedure	factor	unit	denomina	measure
Y	_ALL_	_ALL_	_ALL_	_ALL_	Placebo	count	quantity			86
Y	_ALL_	_ALL_	_ALL_	_ALL_	Placebo	percent	proportion		ittfl	100

Generation of Analysis Results Dataset

– *continued …*

- Convert .csv files to RDF data cubes using the R-package - RRDF interfacing to Apache Jena

```
ds:obs01  a                          qb:Observation ;
        rdfs:comment                 "Statistic for number of records/Statistics for factor with the dimensions XX"@en ;
        rdfs:label                   "1"^^xsd:string ;
        qb:dataSet                   ds:dataset-TAB1X01 ;
        crnd-attribute:denominator   "" ;
        crnd-attribute:unit          "NA"^^xsd:string ;
        crnd-dimension:comp24fl      code:comp24fl-_ALL_ ;
        crnd-dimension:compfl        code:compfl-_ALL_ ;
        crnd-dimension:efffl         code:efffl-_ALL_ ;
        crnd-dimension:factor        code:factor-quantity ;
        crnd-dimension:ittfl         code:ittfl-Y ;
        crnd-dimension:procedure     code:procedure-count ;
        crnd-dimension:saffl         code:saffl-_ALL_ ;
        crnd-dimension:trt01p        code:trt01p-Placebo ;
        crnd-measure:measure         "86"^^xsd:double .
```

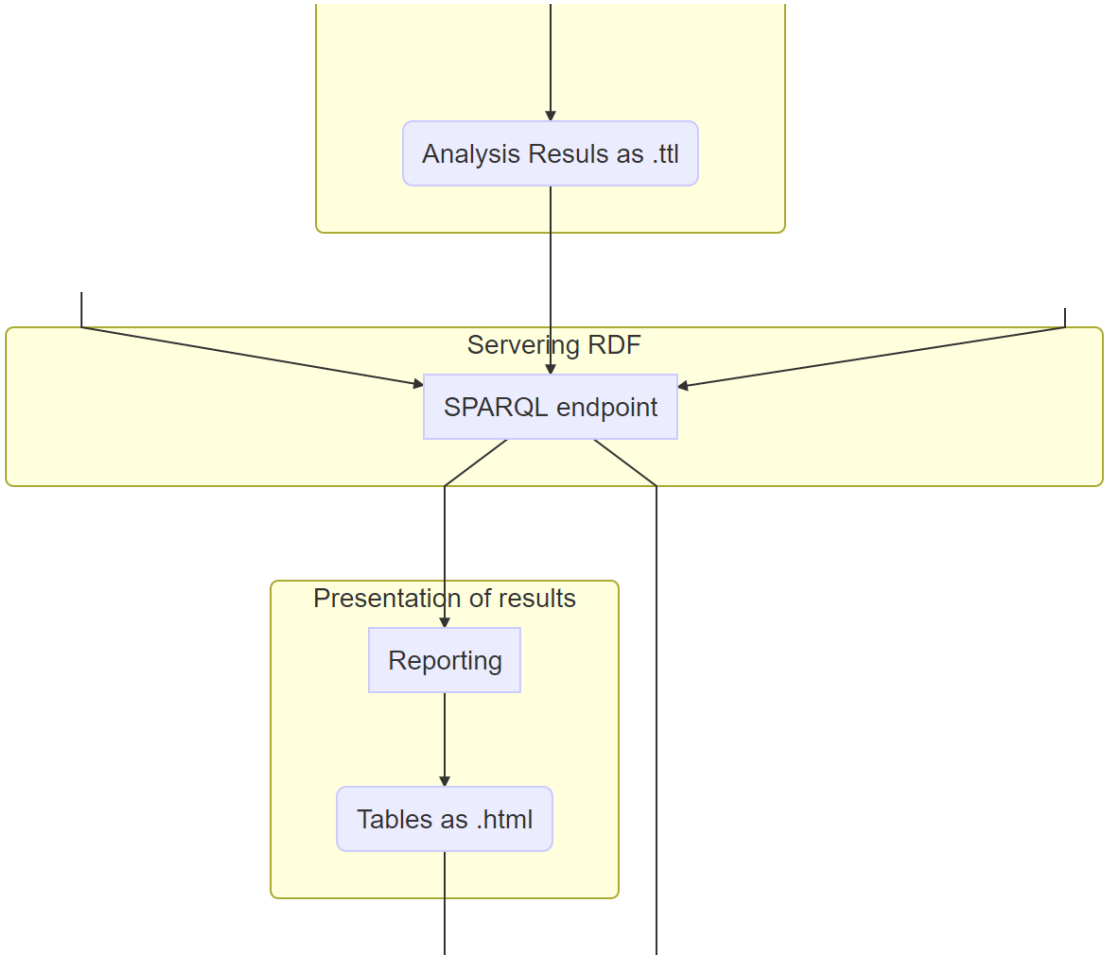
Figure 2: Snapshot of one of the observations from the RDF (.ttl) file showing number of patients in ITT population for placebo group

File naming conventions

Filename	Description
build-tab2x01.cmd	Windows CMD script for generating the outputs
tab2x01.sas	SAS program generating .csv file with results and meta data
TAB2X01.csv	.csv file with the results for the RDF data cube
TAB2X01-Components.csv	.csv file with meta data for the RDF data cube
tab2x01-ttl.Rmd	R script generating RDF data cube using the .csv files
CDISC-pilot-TAB2X01.ttl	The table as RDF data cube
tab2x01-observations.rq	SPARQL SELECT query to get observations for the data cube
tab2x01.rq	SPARQL SELECT query to get table results in format suitable for presentation in SAS
get-tab2x01-with-proc-groovy.sas	SAS program querying RDF data cube and present as HTML with links (href) to cube observations
tab2x01.html	HTML representation of analysis results
File extensions: .cmd – windows cmd script, .sas - SAS system program, .csv – comma separated values, .Rmd - R markdown, .ttl – RDF turtle, .rq – SPARQL query, html – hyper text markup language	

PRESENTATION FROM RDF DATA CUBE

- The generated RDF data cubes are queried using SPARQL (https://www.w3.org/TR/sparql11-overview/)
  - The SPARQL query is performed using a SAS macro
    - PROC GROOVY to interface to Apache Jena (http:// jena.apache.org/), loading the generated RDF file (.ttl) and perform the query, and use Apache Jena to store the results as XML.
    - XML file is processed by the macro using XML mapper to convert it to SAS dataset.
- Tabular output as html/RTF files is created using SAS (PROC REPORT).


Analysis Resuls as .ttl
Servering RDF
SPARQL endpoint
Presentation of results
Reporting
Tables as .html

PRESENTATION FROM RDF DATA CUBE

```
select
 ?ittfl ?procedureZ1 ?col1z1URI ?col1z1
where
{
?col1z1URI  a qb:Observation;
      crnd-dimension:comp24fl ?comp24fl    ;
      crnd-dimension:compfl ?compfl    ;
      crnd-dimension:efffl ?efffl      ;
      crnd-dimension:factor ?factorZ1     ;
      crnd-dimension:ittfl ?ittfl      ;
      crnd-dimension:procedure ?procedureZ1 ;
      crnd-dimension:saffl ?saffl    ;
      crnd-dimension:trt01p code:trt01p-Placebo ;
      crnd-measure:measure ?col1z1 .
 filter (?ittfl = code:ittfl-Y) }
```

Figure 3: Snapshot of SPARQL query that retrieves the observation shown in Figure 4

	ittfl	col1z1URI	col1z1	procedureZ1
1	code:ittfl-Y	<http://www.example.org/dc/tab1x01/ds/obs01>	86	code:procedure-count
2	code:ittfl-Y	<http://www.example.org/dc/tab1x01/ds/obs02>	100	code:procedure-percent

Figure 4: Results produced by the SPARQL query from Figure 3

TRACEABILITY

- Provide reference to a result is to use the URI for the observation

```
<a href="http://www.example.org/rdf-data-cube/obs01">86</a>
```

- Investigate using RDFa to represent citation from an RDF data cube

```
<span vocab="http://www.example.org/citingForCSR/" typeof="ResultReference">
<span property="citeUsingFormat" content="f4.1">
<a property="isFrom" href="http://www.example.org/rdf-data-cube/obs7">
<span property="isShownAs">43.1</span>
</a></span></span>
```

Figure 5: RDFa embedded in HTML referencing an RDF data cube observation

TRACEABILITY

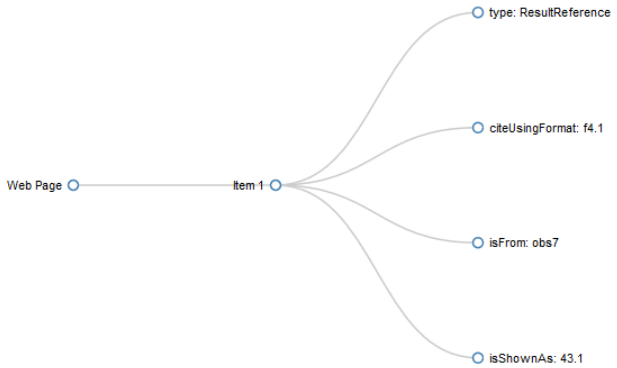


Figure 6: **Visualization of RDFa**

- An RDF data cube observation (figure 2) provides the dimensions for the contributing data.
- For each dimension the RDF data cube codelist contain the original value in the data.

```
@prefix rdfa: <http://www.w3.org/ns/rdfa#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
<http://rdfa.info/play/> rdfa:usesVocabulary <http://www.example.org/
citingForCSR/> .
_:1  rdf:type <http://www.example.org/citingForCSR/ResultReference>;
   <http://www.example.org/citingForCSR/citeUsingFormat> "f4.1";
   <http://www.example.org/citingForCSR/isFrom>
                                  <http://www.example.org/rdf-data-cube/obs7>;
   <http://www.example.org/citingForCSR/isShownAs> "43.1" .
```

Figure 7: RDFa markup as RDF/Turtle

Putting it All Together: Application

- Browser based
- Shows results and performs SPARQL queries
- Generates HTML version of tables and shows linking between ADaM data and tables

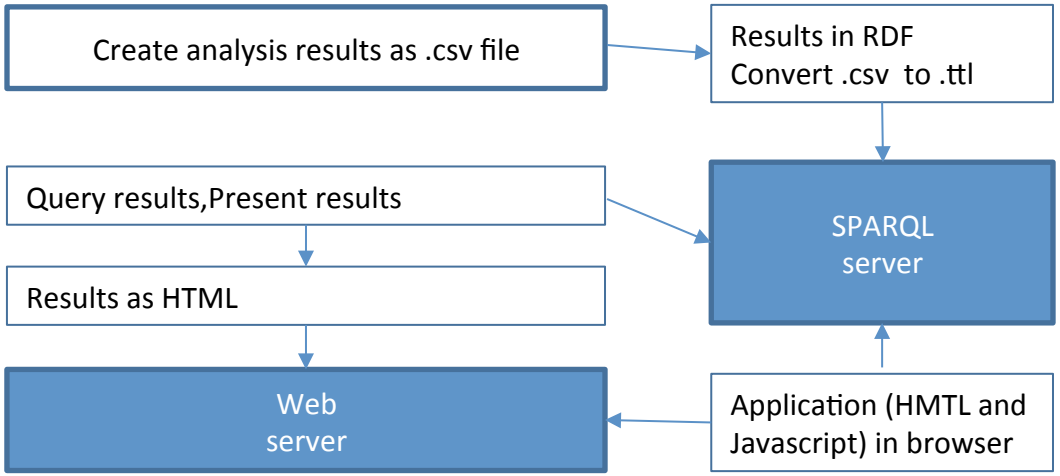
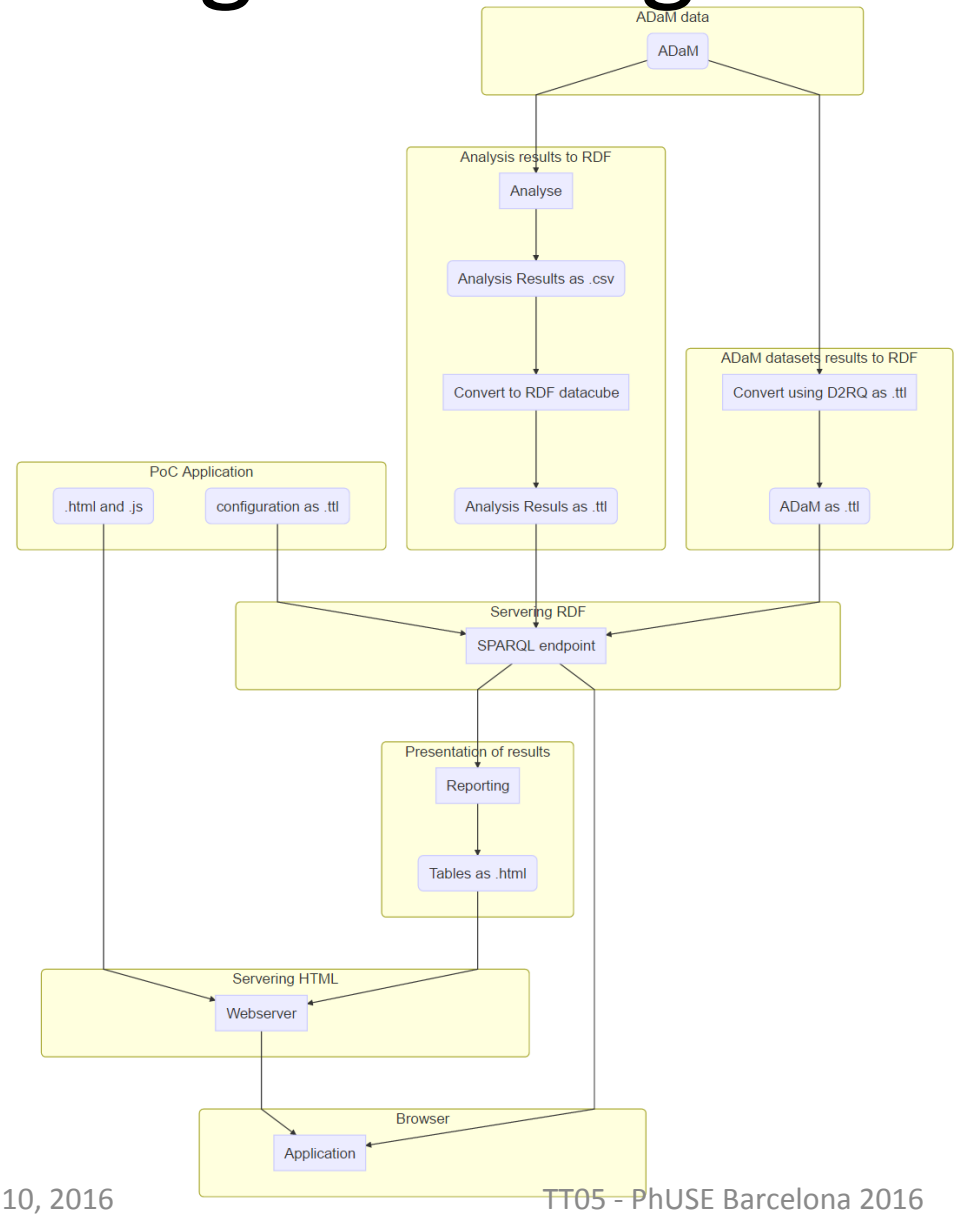


Figure 8: Block diagram showing different components of the application

Putting it All Together: Application



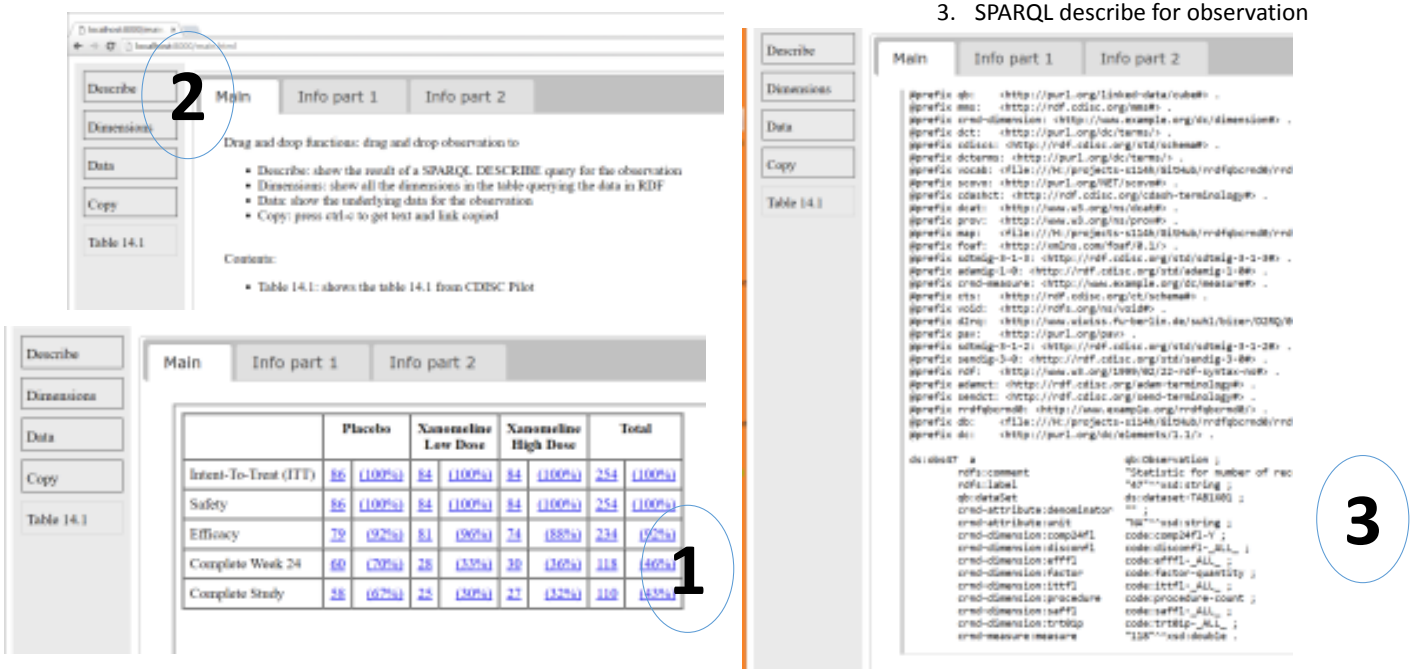
Generated using mermaid using R package DiagrammeR (http://rich-iannone.github.io/DiagrammeR/),

For code, see https://github.com/MarcJAndersen/poc-analysis-results-metadata/blob/master/use-rrdfqbcrnd0/poc-application-overview.Rmd

Putting it All Together: Application

Application

1. Click and hold ”118”
2. Drag to describe
3. SPARQL describe for observation



http://www.phusewiki.org/docs/Conference%202015%20TT%20Papers/TT07.pdf
http://www.phusewiki.org/docs/Conference%202015%20TT%20Presentations/TT07_Dude_Wheres_My_Graph.pptx

Figure 9: Snapshots showing several views of the application
Sourcecode at https://github.com/MarcJAndersen/poc-analysis-results-metadata

Evaluation

- Generation of Results (in CSV)
  - It is feasible to use PROC TABULATE or any other SAS PROC (E.g. LIFETEST) to generate results
  - Handling of missing data/zero counts
- Generation of RDF Data cubes from CSV files
  - R package
  - Challenges in setting up
- Presentation from RDF Data Cube
  - SPARQL queries for data retrieval from RDF
  - Automated generation of SPARQL queries
  - Keep SPARQL queries simple and perform data manipulation in SAS

Conclusion

Overall the potential of using linked data and proposed approach is demonstrated

The following topics could be investigated further

- Use the R tables package (https://cran.r-project.org/web/packages/tables/vignettes/tables.pdf)
- Generate and store metadata for the scripts using the approach from the PhUSE scripting group (https://github.com/phuse-org/phuse-scripts/blob/master/MetaData_template.yml),
- The direct generation of RDF from SAS or R as text files with either SPARQL INSERT or SPARQL CONSTRUCT or as turtle
- Use XSL transformation of RDF/XML for subsequent presentation
- Other methods and tools for presenting RDF (for example Dokeieli, https://github.com/linkeddata/dokieli).

Conclusion *continued….*

The following topics could be investigated further:

- Use proposed approach to create figures
- Hyperlinking results in CSR body of text
- Suggest format for analysis results as a CDISC standard (like ADaM specification)
- Suggest standard for representing DEFINE-xml as RDF
- Suggest RDF representation of ADaM datasets connecting to CDISC standards in RDF

Thank You

Questions?