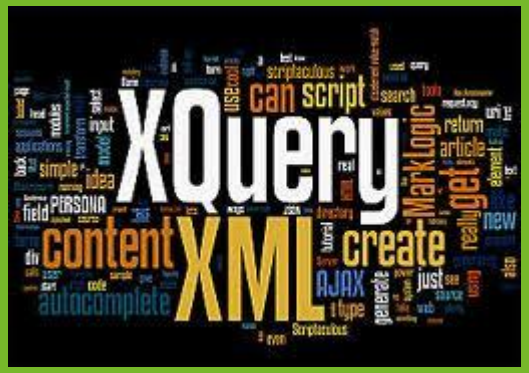
SDTM Validation Rules in XQuery



FH-Prof. Dr. Jozef Aerts
Institute for eHealth
Univ. Appl. Sciences FH Joanneum
Graz, Austria

Can you understand the following validation rule (part 1)?

```
1  (: Rule FDAC040 - Subject is not present in DM domain - All Subjects (USUBJID) must be present in Demographics (DM)
   domain :)
2  xquery version "3.0";
3  declare namespace def = "http://www.cdisc.org/ns/def/v2.0";
4  declare namespace odm="http://www.cdisc.org/ns/odm/v1.3";
5  declare namespace data="http://www.cdisc.org/ns/Dataset-XML/v1.0";
6  declare namespace xlink="http://www.w3.org/1999/xlink";
7  let $base := '/db/fda_submissions/cdisc01/'
8  let $define := 'define2-0-0-example-sdtm.xml'
9  let $domain := 'CM'
10 (: we need the ItemOID of the USUBJID variable - and need to take care of the use case that people have used
   different ItemDefs for the same variable in different domains/datasets :)
11 (: first get the one for the DM dataset :)
12 (: let $dmitemgroupdef := doc(concat($base,$define)) :)
13 let $dmitemgroupdef := doc(concat($base,$define))//odm:ItemGroupDef[@Name='DM']
14 let $dmdatasetname := $dmitemgroupdef/def:leaf/@xlink:href
15 let $dmdatasetpath := concat($base,$dmdatasetname)
16 (: EITHER provide $domain=:'ALL', meaning: validate for all domains referenced from the define.xml OR:
17 $domain:='XX' where XX is a specific domain, MEANING validate for a single domain only :)
18 (:  get the definitions for the domains (ItemGroupDefs in define.xml) :)
19 let $domains := (
20     if($domain != 'ALL') then doc(concat($base,$define))//odm:ItemGroupDef[@Domain=$domain]
21     else doc(concat($base,$define))//odm:ItemGroupDef
22 )
```

Can you understand the following validation rule (part 2)?

```
22  )
23  (: Get the OID of USUBJID in DM :)
24  let $usubjoiddm := (
25      for $a in doc(concat($base,$define))//odm:ItemDef[@Name='USUBJID']/@OID
26      where $a = doc(concat($base,$define))//odm:ItemGroupDef[@Name='DM']/odm:ItemRef/@ItemOID
27      return $a
28  )
29  (: now iterate over all dataset definitions in the define.xm and get the USUBJID :)
30  for $itemgroupdef in $domains
31      let $dataset := $itemgroupdef/def:leaf/@xlink:href
32      let $datasetpath := concat($base,$dataset)
33      (: find the variable for which the name is 'USUBJID' :)
34      let $usubjidoid := (
35          for $a in doc(concat($base,$define))//odm:ItemDef[@Name='USUBJID']/@OID
36          where $a = $itemgroupdef/odm:ItemRef/@ItemOID
37          return $a
38      )
39      for $d in doc($datasetpath)//odm:ItemData[@ItemOID=$usubjidoid]
40          let $recnum := $d/../@data:ItemGroupDataSeq
41          let $value := $d/@Value
42              (: get the ones for which no value in the DM dataset is found :)
43              where not(doc($dmdatasetpath)//odm:ItemData[@ItemOID=$usubjoiddm][@Value=$value])
44              return <error rule="FDAC040" rulelastupdate="2015-09-08" recordnumber="{data($recnum)}">USUBJID {data
    ($value)} in dataset {data($dataset)} could not be found in DM dataset</error>
45
```

The problem we want to tackle

- (SDTM) validation rules are usually published:
  - As pure text
  - in Excel worksheets
  - In non-machine-readable/executable code
  - open for different interpretation

Example of an FDA SDTM validation rule

Rule: FDAC068:

Records for subjects who failed a screening or were not assigned to study treatment (ARMCD is 'SCRNFAIL' or 'NOTASSGN') should not be included in the Trial Arms (TA) or Trial Visits (TV) datasets

What is meant here?

Implementation of CDISC/FDA validation rules

- Usually in software (open-source or not)
- Own interpretation of the implementors
- Intransparent (or you need to dig into the source code)
- Often weird implementations
  - E.g. leading to many false poitives
  - But intransparent how they were really implemented

An alternative

- Why not write the rules in a language that
  - Is human readable and understandable (by usual SDTM/ADaM/SEND specialist)
  - Is machine-executable

- Such a language is XQuery
  - XQuery = „XML Query Language“
  - So essentially for XML data

Disadvantages

- Mainly for quering XML files – forget about SAS Transport 5
- Slower – queries must first be compiled
- XQuery is not software: you need a software to execute the queries
  (like MySQLWorkbench for relational DB)
- Yet another technology …

- But we now have Define.xml and Dataset-XML isn't it?

Principles

- Define.xml is leading
  - Tells us where the submission files are
  - Gives us the information about data types, lengths, enumerations
  - Provides the codelists

- Your define.xml needs to correctly describe your submission!

A simple rule in XQuery

```
for $itemgroupdef in $domains
let $dataset := $itemgroupdef/def:leaf/@xlink:href
let $datasetpath := concat($base,$dataset)
(: find the variable for which the name is 'USUBJID' :)
let $usubjidoid := (
for $a in doc(concat($base,$define))//odm:ItemDef[@Name='USUBJID']/@OID
where $a = $itemgroupdef/odm:ItemRef/@ItemOID
return $a
)
for $d in doc($datasetpath)//odm:ItemData[@ItemOID=$usubjidoid]
let $recnum := $d/../@data:ItemGroupDataSeq
let $value := $d/@Value
(: get the ones for which no value in the DM dataset is found :)
where
not(doc($dmdatasetpath)//odm:ItemData[@ItemOID=$usubjoiddm][@Value=$value])
return <error rule="FDAC040" rulelastupdate="2015-09-08"
recordnumber="{data($recnum)}">USUBJID {data($value)} in dataset {data($dataset)}
could not be found in DM dataset</error>
```

What has been done sofar?

- 90% of all FDA-SDTM rules were written as XQuery
- Except for
  - Those that are nonsense, wrong, are an expectation rather than a rule
  - Those that needs MedDRA lookup
    - License needed

http://cdiscguru.blogspot.com/2015/02/rule-fdac084-is-just-damned-wrong.html

Where can I get it

- http://xml4pharmaserver.com/WebServices/XQueryRules_webservices.html

- A web service is available to retrieve them
  - By ID (e.g. „FDAC091“)
  - By class or domain
  - By last update
  - By Standard, Originator, … (to come)

How to work with them?

- Only for define.xml and Dataset-XML files
- In a file system (slower) or using a native XML database (eXist, BaseX, …)
- You will need an XQuery engine, e.g. „eXide“ (part of eXist - http://www.exist-db.org)
- Or write your own software (example provided on the website)

Example running Xquery using eXist / eXide

```
1  (: Rule FDAC040 - Subject is not present in DM domain - All Subjects (USUBJID) must be present in Demographics
   (DM) domain :)
2  xquery version "3.0";
3  declare namespace def = "http://www.cdisc.org/ns/def/v2.0";
4  declare namespace odm="http://www.cdisc.org/ns/odm/v1.3";
5  declare namespace data="http://www.cdisc.org/ns/Dataset-XML/v1.0";
6  declare namespace xlink="http:
7  let $base := '/db/fda_submissi
8  let $define := 'define2-0-0-ex
9  let $domain := 'CM'
10 (: we need the ItemOID of the
   different ItemDefs for the sam
11 (: first get the one for the D
12 let $dmitemgroupdef := doc(con
13 let $dmdatasetname := $dmitemg
14 let $dmdatasetpath := concat($
15 (: EITHER provide $domain:='AL
   :='XX' where XX is a specific
16 (:  get the definitions for th
17 let $domains := (
18     if($domain != 'ALL') then
19     else doc(concat($base,$define))//odm:ItemGroupDef
20 )
21 (: Get the OID of USUBJID in DM :)
22 let $usubjoiddm := (
23     for $a in doc(concat($base,$define))//odm:ItemDef[@Name='USUBJID']/@OID
```

XML Output | Live Preview

```
<error rule="FDAC040"
rulelastupdate="2015-09-08"
recordnumber="3">USUBJID
INVALID_CDISC01.100008 in
dataset cm.xml could not be
found in DM dataset</error>
```

What‘s next?

- CDISC SDTM validation rules (SDTM Validation Subteam) are being implemented
- Anyone wanting to do the ADaM rules?
- SEND rules?

- Make all rules publicly available using the website & webservice
  - No need to „wait for the next release“

Long term goals

- Rules development based on consensus within the SDTM community
- Fully transparent implementation
- Governed by CDISC volunteers (not by a company)
- Building a „real open source“ community

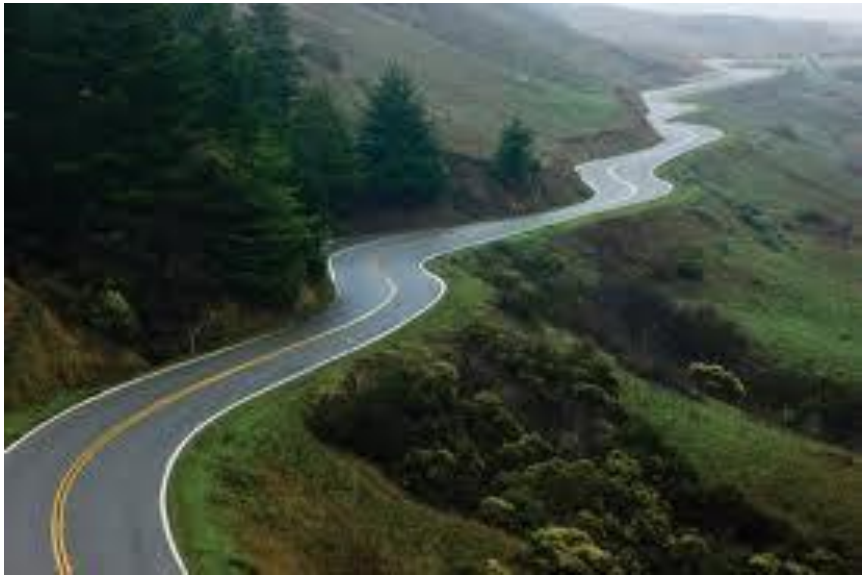
for rules development

Long term goals

- eSHARE will get an API in the future
- eSHARE is thinking about establishing (RESTful) web services
  - e.g. answering questions like „is value X a valid coded value for variable Y?“
- Validation rules in XQuery are planned to become part of eSHARE

The end?

- I don't think so …



The long and winding road to interoperability …