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)?

```xquery
(: Rule FDAC040 - Subject is not present in DM domain - All Subjects (USUBJID) must be present in Demographics (DM) domain :)

xquery version "3.0";
declare namespace def = "http://www.cdisc.org/ns/def/v2.0";
declare namespace odm="http://www.cdisc.org/ns/odm/v1.3";
declare namespace data="http://www.cdisc.org/ns/Dataset-XML/v1.0";
declare namespace xlink="http://www.w3.org/1999/xlink";
let $base := '/db/fda_submissions/cdisc01/
let $define := 'define2-0-0-example-sdtm.xml'
let $domain := 'CM'
(: 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 :)
(: first get the one for the DM dataset :)
let $dmitemgroupdef := doc(concat($base,$define)) :
let $dmitemgroupdef := doc(concat($base,$define))//edm:ItemGroupDef[@Name='DM']
let $dmdatasetname := $dmitemgroupdef/def:leaf/@xlink:href
let $dmdatasetpath := concat($base,$dmdatasetname)
(: EITHER provide $domain='ALL', meaning: validate for all domains referenced from the define.xml OR: $domain='XX' where XX is a specific domain, MEANING validate for a single domain only :) 
(: get the definitions for the domains (ItemGroupDefs in define.xml) :)
let $domains := {
    if($domain != 'ALL') then doc(concat($base,$define))//edm:ItemGroupDef[@Domain=$domain]
    else doc(concat($base,$define))//edm:ItemGroupDef
}
```
Can you understand the following validation rule (part 2)?

```xml
22 )
23 (: Get the OID of USUBJID in DM :)  
24 let $usubjiddm := {
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 := $d/@Value
42     (: get the ones for which no value in the DM dataset is found :)  
43     where not(doc($datasetpath)/odm:ItemData[@ItemOID=$usubjidoid][@Value=$value])
44     return <error rule="PDAC040" rulelastupdate="2015-09-08" recordnumber="(data($recnum))">USUBJID {data ($value}) in dataset {data($dataset}) could not be found in DM dataset</error>
```
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*
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 positives
  • 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 querying 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

```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 so far?

• 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

- [Link](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

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

SDTM Validation Rules in XQuery

Jozef Aerts
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 ...