CDISC 2019 Europe Interchange

Amsterdam, Netherlands | 6-10 May

Open Rules for CDISC.
Web Services and the CDISC Library API

Presented by Jozef Aerts
XML4Pharma

2019-05-08

2019 Europe Interchange

Amsterdam, Netherlands | 6-10 May 2019

Agenda

1. What is wrong with current validation of submission data?
2. The alternative: "Open Rules for CDISC Standards"
3. Validation Rules and the CDISC Library API

Disclaimer:
The views and opinions expressed in this presentation are those of the Jozef Aerts and do not necessarily reflect the official policy or position of CDISC.

What is wrong with current validation of submission data?

- When publishing Implementation Guides (SDTM / SEND / ADaM), CDISC did not publish corresponding validation rules in the past (or only *years after* the publication of the IG)

- Implementation Guides are (very) open for different interpretations
  - Published as PDF or HTML (not machine-readable)
  - Vague language - e.g. *"The following Qualifiers would not generally be used in XX …"*

- Other organizations have picked up this lack of clarity to develop validation rules and impose these as "the gold standard"

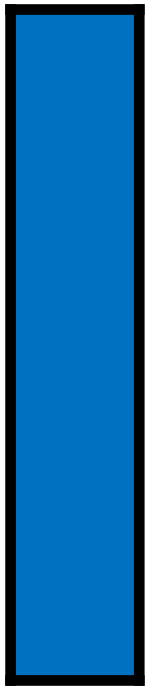
- These "rules" have been extended by / on behalf of FDA and PMDA, becoming the "FDA/PMDA Validation Rules"

What is wrong with current validation of submission data?

- Clarity can only be obtained when CDISC defines the validation rules
- And makes them available in a machine-readable format
  - So that different interpretations become impossible (no "wiggle room")
  - Publishing them in Excel is just a first step, but not sufficient
- All new CDISC standards will publish validation rules

Validation of Submission Data - Why we want to do better
Current validation software …

- Is not governed nor maintained by CDISC
- Is often an over-interpretation of the IGs
- Rule implementations are intransparent
- Some (FDA/PMDA) rules do not make sense
  (e.g. --ORRESU must be populated when --ORRES populated)
- Mostly ignores the define.xml (define.xml = "sponsor's truth")
- Uses manipulated (!) CDISC-CT files
- Generates a large number of false positives
- Has a lot of bugs
- Does not have a mechanism for fast bug fixes
  - Bug fixes will come in 2, 3, 4, … years
- Bugs and false positives are to be reported in the SDRG
- The world upside down …



30 seconds overview

CDISC creates clarity

How can we bring more clarity in validation?

Open Rules for CDISC Standards

Open Rules for CDISC Standards

Requirements

Requirements for Open Rules for CDISC Standards

- Vendor neutral, freely available
- Machine-readable AND human-readable - transparent
- Separation of rules and software
  - Rules can be read and executed from within **any** (modern) software
- Takes define.xml serious as "the sponsor's truth"
- Reference implementation available
- Transformable to different languages (R, SAS, Java, Python, …)
- No false positives
- Timely and transparent updates when bugs reported
- Process for making updates and fixes automatically available
  - E.g. API and RESTful Web Service
- Versioning of rules
- Support for SDTM, SEND and ADaM
- Easily extensible by sponsors with company-internal rules

Coming to a vendor-neutral, human- and machine-readable rule description

- Unfortunately, there does not seem to be a universal, open, machine-readable language for validation rules
- **Candidates** currently investigated:
- XQuery,
  - Which is however limited to XML
  - But we DO have CDISC Dataset-XML for SDTM, SEND and ADaM!
  - International W3C standard
  - Easy-to-learn and could be transformed into R, SAS, Python …
- VTL: Validation and Transformation Language
  - Seems to be pretty abstract, little experience available
- KNIME: Data mining software and language

Example: a simple SDTM Rule (from CDISC)

CG0027: The values of --CAT and --SCAT must be different

	A	B	C	D	E	F	G	H	I	J
22	CG0026	ALL	ALL	--RFTDTC	--RFTDTC = null	--TPTREF = null	Model v1.4	2.2.5	Table 2.2.5: --RFTDTC	Date/time for a fixed reference time point defined by --TPTREF
23	CG0027	ALL	ALL	--SCAT	--SCAT ^= --CAT	--SCAT ^= null	Model v1.4	2.2.1	Table 2.2.1: --SCAT	Used to define a further categorization of --CAT values.
	CG0028	ALL	ALL	--SEQ	--SEQ is a unique number per USUBJID per domain, or a unique number per POOLID	Variable is not TS.TSSEQ	Model v1.4	2.2.4	Table 2.2.4: --SEQ	Sequence number to ensure uniqueness of records within a dataset for a subject (or within

- Excel worksheet gives precondition, condition, rule
- Is pretty well human-readable
- But hardly machine-executable
  - And can, in some cases, be interpreted differently by humans

Open Rules for CDISC Standards

Development and Implementation

A simple rule in XQuery (shortened):
CG0027: The values of --CAT and --SCAT must be different

```
(: iterate over all datasets that have a --CAT and --SCAT variable in the interventions, events findings domains and the TI dataset :)
for $dataset in
    doc(concat($base,$define))//odm:ItemGroupDef[upper-case(@def:Class)='INTERVENTIONS' or upper-case(@def:Class)='EVENTS' or upper-ca
    let $name := $dataset/@Name
    let $datasetname := $dataset/def:leaf/@xlink:href
    let $datasetlocation := concat($base,$datasetname)
    (: Get the OID of the --CAT variable (if any) :)
    let $catoid := (
        for $a in doc(concat($base,$define))//odm:ItemDef[ends-with(@Name,'CAT') and string-length(@Name)=5]/@OID
        where $a = doc(concat($base,$define))//odm:ItemGroupDef[@Name=$name]/odm:ItemRef/@ItemOID
        return $a
    )
    let $catname := concat($name,'CAT')
    (: Get the OID of the --SCAT variable (if any) :)
    let $scatoid := (
        for $a in doc(concat($base,$define))//odm:ItemDef[ends-with(@Name,'SCAT') and string-length(@Name)=6]/@OID
        where $a = doc(concat($base,$define))//odm:ItemGroupDef[@Name=$name]/odm:ItemRef/@ItemOID
        return $a
    )
    let $scatname := concat($name,'SCAT')
    (: iterate over all records that DO have a CAT AND SCAT value :)
```

Selects the datasets that have both --CAT as well as --SCAT as per the define.xml

A simple rule in XQuery (shortened):
CG0027: The values of --CAT and --SCAT must be different

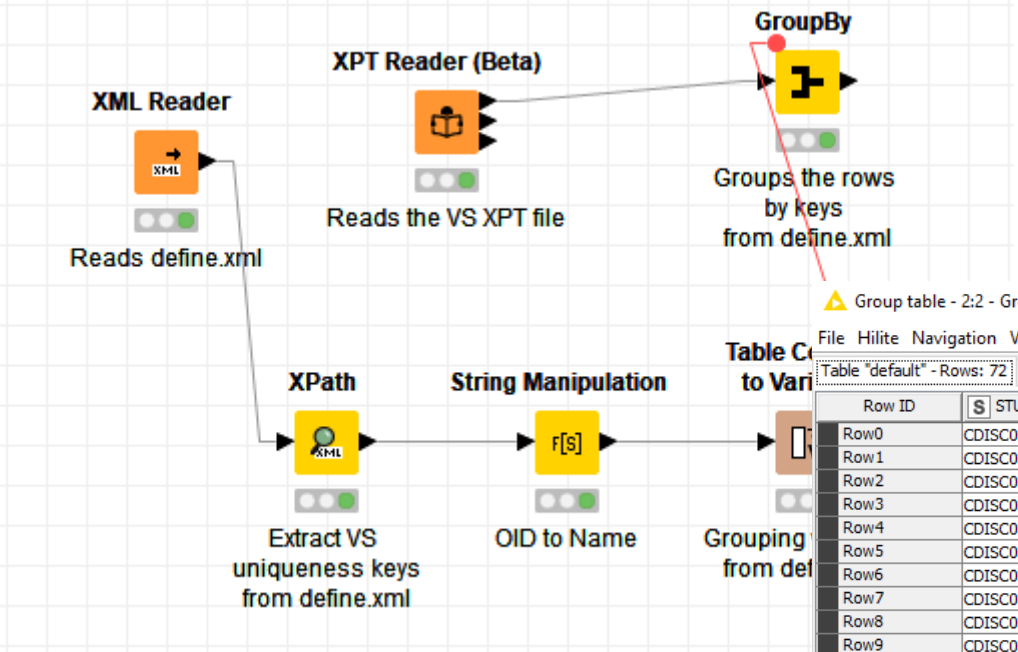
```
33      (: iterate over all records that DO have a CAT AND SCAT value :)
34      for $record in doc($datasetlocation)//odm:ItemGroupData[odm:ItemData[@ItemOID=$catoid]
35              and odm:ItemData[@ItemOID=$scatoid]]
36          let $recnum := $record/@data:ItemGroupDataSeq
37          (: and compare the values - they MUST NOT be equal :)
38          let $catvalue := $record/odm:ItemData[@ItemOID=$catoid]/@Value
39          let $scatvalue := $record/odm:ItemData[@ItemOID=$scatoid]/@Value
40          where $catvalue = $scatvalue
41          return <error rule="CG0027" variable="{data($scatname)}" dataset="{$name}"
42              rulelastupdate="2017-02-04" recordnumber="{data($recnum)}">Values for {data($catname)}
43              and {data($scatname)} are identical in dataset {data($datasetname)}</error>
```

And iterates over all records in the selected datasets
and compares the values of --CAT and --SCAT.
Error messages come as XML (transformable to HTML, PDF, … reports)

This code is very easy to read and understand,
even for SAS programmers 

Another simple rule in KNIME:

CG0019: Each record is unique per sponsor defined key variables as documented in the define.xml



Open for Innovation ®
KNIME

Group table - 2:2 - GroupBy

File Hilite Navigation View

Table "default" - Rows: 72 | Spec - Columns: 7 | Properties | Flow Variables

Row ID	STUDYID	USUBJID	VSTEST...	VSPOS	VISITNUM	VSDTC	Count*...
Row0	CDISC01	CDISC01.100008	DIABP	SITTING	1	2003-04-15	1
Row1	CDISC01	CDISC01.100008	DIABP	SITTING	2	2003-04-29	1
Row2	CDISC01	CDISC01.100008	DIABP	SITTING	3	2003-05-13	1
Row3	CDISC01	CDISC01.100008	DIABP	SITTING	10	2003-10-13	1
Row4	CDISC01	CDISC01.100008	FRMSIZE	?	1	2003-04-15	1
Row5	CDISC01	CDISC01.100008	HEIGHT	?	1	2003-04-15	1
Row6	CDISC01	CDISC01.100008	PULSE	?	1	2003-04-15	1
Row7	CDISC01	CDISC01.100008	PULSE	?	2	2003-04-29	1
Row8	CDISC01	CDISC01.100008	PULSE	?	3	2003-05-13	1
Row9	CDISC01	CDISC01.100008	PULSE	?	10	2003-10-13	1
Row10	CDISC01	CDISC01.100008	SYSBP	SITTING	1	2003-04-15	1
Row11	CDISC01	CDISC01.100008	SYSBP	SITTING	2	2003-04-29	1
Row12	CDISC01	CDISC01.100008	SYSBP	SITTING	3	2003-05-13	1
Row13	CDISC01	CDISC01.100008	SYSBP	SITTING	10	2003-10-13	1
Row14	CDISC01	CDISC01.100008	WEIGHT	?	1	2003-04-15	1
Row15	CDISC01	CDISC01.100008	WEIGHT	?	2	2003-04-29	1
Row16	CDISC01	CDISC01.100008	WEIGHT	?	3	2003-05-13	1
Row17	CDISC01	CDISC01.100008	WEIGHT	?	10	2003-10-13	1
Row18	CDISC01	CDISC01.100014	DIABP	SITTING	1	2003-10-06	1

Who will be able to implement these Open Rules?

- Everyone using modern computer languages
- Software is strictly separated from the rules themselves
  - But any modern software will be able to read and execute the rules
- Includes SAS, R, Java, Python, C#, …
- Will usually only require a few lines of code


```
XQDataSource xqjd = new SaxonXQDataSource(new net.sf.saxon.Configuration());
XQConnection xqjc = xqjd.getConnection();
XQPreparedExpression exp = xqjc.prepareExpression(xqueryString);
exp.bindObject(new QName("base"), base, null);
XQResultSequence result = exp.executeQuery();
while (result.next()) { … }
```

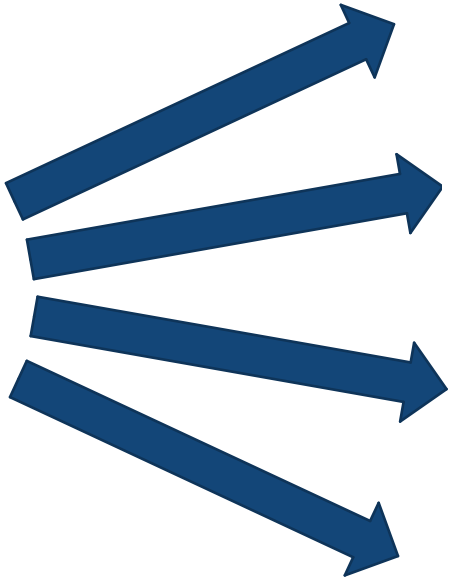
What is a "Reference Implementation"?

- Well known in the software development world
- Is an implementation of an *OPEN STANDARD* that serves as an example implementation
- Usually the Reference Implementation is *FULLY OPEN SOURCE*
  - Including the right to copy, change code, redistribute source and executables
- Everyone is *COMPLETELY FREE* to develop other implementations
  - Including commercial implementations
  - Competition is based on execution speed, user-friendlyness, additional features
- BUT! The results *MUST* be the same as when using the reference implementation

Generating other implementations from the Reference Implementation

Open Rules for CDISC Standards

Reference Implementation



Java source code

SAS source code

R source code

Python source code

transport formats

XPT

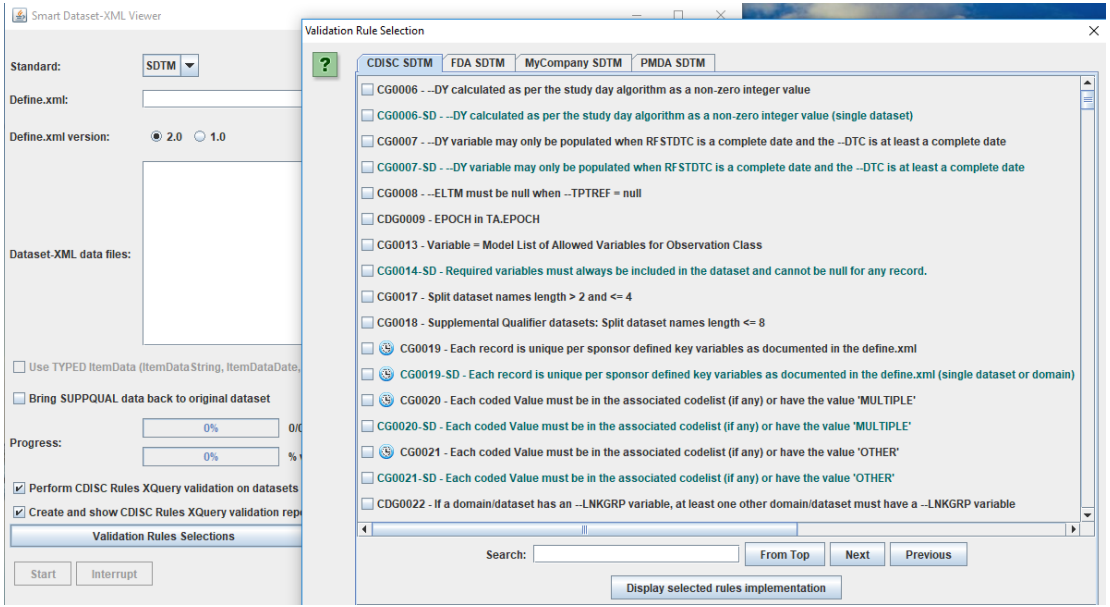
XML

RDF

JSON

Current (testbed) Implementations

- The open source "Smart Submission Dataset Viewer" already implements a good amount of such "candidate" "Open Rules" (including FDA, PMDA)



https://sourceforge.net/projects/smart-submission-dataset-viewe/

Open Rules and RESTful web services

- Some of the "candidate" "open rules" do use (query) RESTful webservices
  - From NLM (e.g. validity of LOINC codes), NIH (e.g. RxCUI/RxNorm) and CDISC-CT
  - Avoids needing to update files constantly (such as CT files)
  - Will start using CDISC Library RESTful web services (former SHARE 2.0) soon
- Currently, "candidate" rules are available from a RESTful web service itself
  - Later part of the CDISC Library?
  - Your application can automatically query whether it has the latest version of the rule
  - Bug fixes in the rules can thus be made available timely
- Sponsor defined rules can easily be added and query the sponsor's own RESTful intranet services or the CDISC Library

Open Rules and the CDISC Library API

- The CDISC Library API describes a number of RESTful web services that can be used by every member
- Open Rules will start using the CDISC Library API as soon as available
  - But will then require a CDISC Library license
- The Open Rules themselves may become available through the CDISC Library API
  - So that you (or better: your application) can always automatically retrieve the latest version - without the need of any software update

Open Rules and the CDISC Library API - Example (possible future implementation)

- For each variable from your define.xml … the query …
  - Asks the CDISC Library whether there is an associated codelist
  - Checks the define.xml codelist against the one from the CDISC Library (taking CT-version and extensibility into account)
  - Checks whether also "OTHER", "MULTIPLE", … is allowed for that variable
    from the CDISC Library
  - Validates the contents in the submission files against the allowed values from the CDISC Library

Open Rules for CDISC Standards
Current actions

- "Open Rules for CDISC Standards" is now a formal CDISC Team
- Ongoing discussions with the "SDTM Rules" and "ADaM Rules" teams to further implement the rules (as defined by the SDTM and ADaM teams)
- And with CDISC management to decide what kind of implementation may become the "Reference Implementation"
- Starting discussions with FDA and PMDA to implement the FDA and PMDA rules as "Open Rules for CDISC Standards", and to make them the FDA and PMDA "reference implementation"

CDISC creates clarity

Thank You!

E-Mail: Jozef.Aerts@XML4Pharma.com

Movie "Implementing the ADaM Rule '*A variable with a suffix of FL has a value that is not Y, N or null*' " in XQuery
http://xml4pharmaserver.com/OpenRulesForCDISC-Standards/movies/XQuery_ADaM_Rule_5.mp4

cdisc