



2026 Europe Interchange

THE FUTURE IS CONNECTED: STANDARDS AND AI POWERING DIGITAL TRANSFORMATION



MILAN, ITALY | MAIN CONFERENCE: 20-21 MAY | TRAININGS & WORKSHOPS: 18, 19, & 22 MAY

Developing Custom Rules for CDISC-CORE

Jozef Aerts
XML4Pharma



Agenda

- CORE Implementation
- Developing Custom Rules
- Generating CORE Rules from "SDTM Dataset Specializations"
- Summary





CORE Implementation



CORE Implementation

- CORE is now well established (**25k downloads**) as the ultimate (base) tool for validation of SDTM and SEND datasets against the CDISC and FDA Business Rules.
- high **quality** of the rules
- full **transparency** and fully **open source** availability, including how the rules have been implemented (as opposed to "black box")
- also **suited for use during mapping** development
- companies can develop/add **"custom" rules**

Companies started building GUIs on top of CORE

- Mapping tools have incorporated CORE, including a GUI
 - CORE can already be used DURING the mapping effort
- Open Source "Smart Submission Dataset Viewer"
- Bioforum Submit 24
- Appsilon Web CORE implementation
- Several others ...

Smart Submission Dataset Viewer

Standard: SDTM ▼ Options

Dataset source type: ☒ Dataset-JSON 1.1 ☐ Dataset-NDJSON ☐ CSV Files

Dataset-JSON metadata: ☒ Use define.xml for metadata ☐ Use Dataset-JSON internal metadata

Define.xml: D:\MetaDataSubmissionGuide_2_0_JSON_1-1_Files_testing\SDTM_Dataset-JSON_with_violations\define.xml Browse Get from API

Define.xml version: ☒ 2.1 ☐ 2.0 ☐ 1.0 View

Dataset-JSON data files:

D:\MetaDataSubmissionGuide_2_0_JSON_1-1_Files_testing\SDTM_Dataset-JSON_with_violations\define.xml
D:\MetaDataSubmissionGuide_2_0_JSON_1-1_Files_testing\SDTM_Dataset-JSON_with_violations\define.xml
D:\MetaDataSubmissionGuide_2_0_JSON_1-1_Files_testing\SDTM_Dataset-JSON_with_violations\define.xml
D:\MetaDataSubmissionGuide_2_0_JSON_1-1_Files_testing\SDTM_Dataset-JSON_with_violations\define.xml
D:\MetaDataSubmissionGuide_2_0_JSON_1-1_Files_testing\SDTM_Dataset-JSON_with_violations\define.xml
D:\MetaDataSubmissionGuide_2_0_JSON_1-1_Files_testing\SDTM_Dataset-JSON_with_violations\define.xml
D:\MetaDataSubmissionGuide_2_0_JSON_1-1_Files_testing\SDTM_Dataset-JSON_with_violations\define.xml
D:\MetaDataSubmissionGuide_2_0_JSON_1-1_Files_testing\SDTM_Dataset-JSON_with_violations\define.xml
D:\MetaDataSubmissionGuide_2_0_JSON_1-1_Files_testing\SDTM_Dataset-JSON_with_violations\define.xml
D:\MetaDataSubmissionGuide_2_0_JSON_1-1_Files_testing\SDTM_Dataset-JSON_with_violations\define.xml

☒ Show record number in first column
☒ Bring SUPPQUAL data back to original dataset

Progress:

100%	23/29 files read - current: relrec_studyID_corrected.json
0%	% validation done
0%	CDISC Library

☒ Perform CORE validation on datasets ☒ Show CORE violations in dataset table in viewer ☐ Only perform CORE validation DO NOT display datasets themselves

☒ Create and show Validation report table ☐ Also generate Excel report

MedDRA Files Directory WHODrug Files Directory

CORE validation progress: 0%

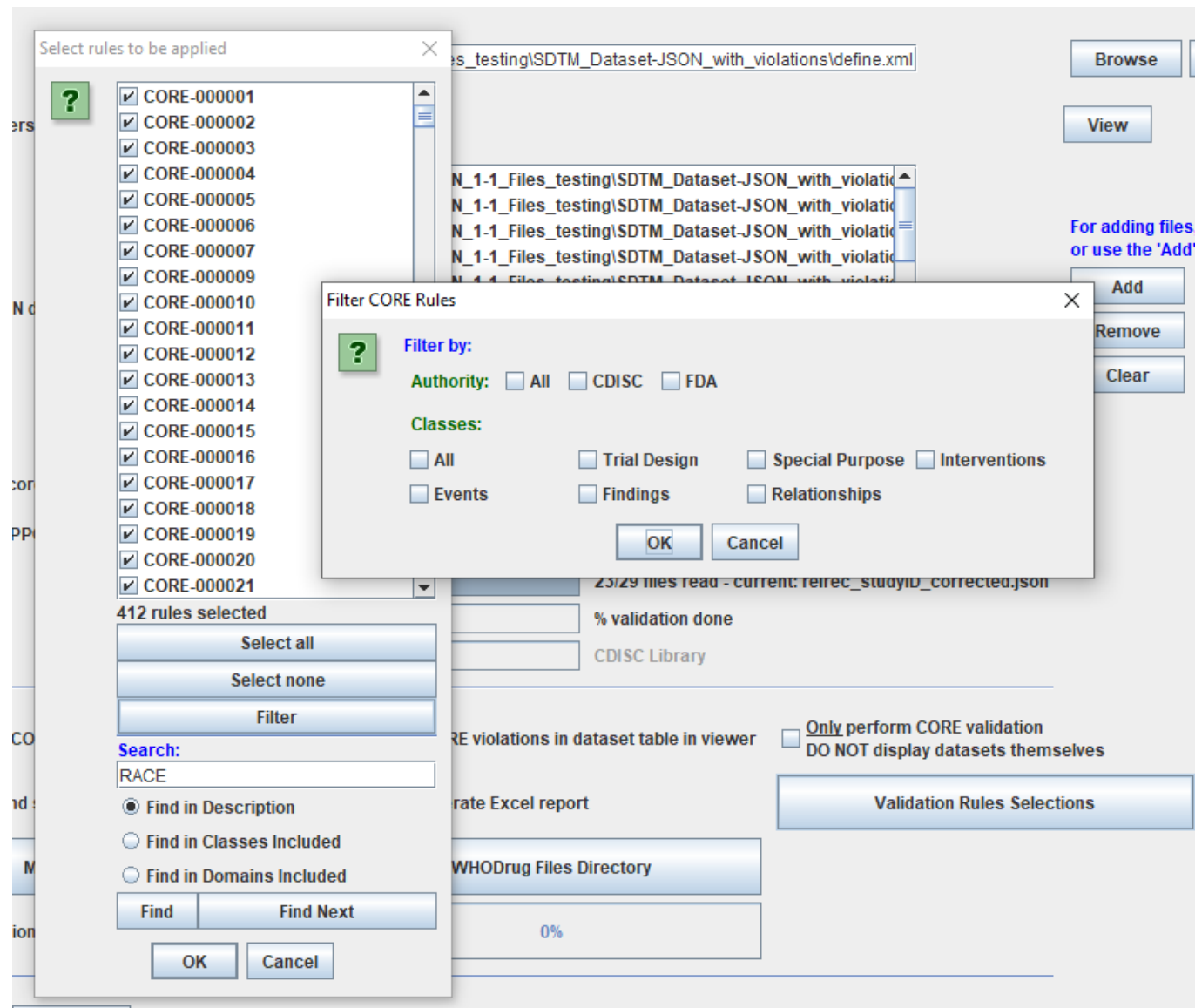
Start Interrupt

Add Add from API
Remove
Clear

For adding files, you can use drag-and-drop, or use the 'Add' button

Validation Rules Selections

Open Source "Smart Submission Dataset Viewer"



Open Source "Smart Submission Dataset Viewer"

DOMAIN	USUBJID	SUBJID	RFSTDTC	RFENDTC	RFXSTDTC	RFXENDTC	RFICDTC	RFPENDTC	DTHDTC	DT
DM	CDISC001	1115	2012-11-30	2013-01-23	2012-11-30	2013-01-23	2012-11-23	2013-05-20		
DM	CDISC002	1211	2012-11-15	2013-01-14	2012-11-15	2013-01-12	2012-10-30	2013-01-14	2013-01-14	Y
DM	CDISC001	1302	2013-08-29	2013-11-05	2013-08-29	2013-11-05	2013-08-20	2014-02-13		
DM	CDISC004	1345	2013-10-08	2014-03-18	2013-10-08	2014-03-18	2013-10-01	2014-03-18		
DM	CDISC005	1383	2013-02-04	2013-08-06	CORE violation: CORE-000239: RFXSTDTC does not equal the earliest value of EX.EXSTDTC (RFXSTDTC)					
DM	CDISC006	1429	2013-03-19	2013-04-30						
DM	CDISC007	1444	2013-01-05	2013-02-13	2013-01-05	2013-02-12	2012-12-31	2013-06-20		
DM	CDISC008	1445	2014-05-11	2014-11-01	2014-05-11	2014-11-01	2014-05-01	2014-11-01	2014-11-01	Y
DM	CDISC009	1087	2012-10-22	2013-04-28	2012-10-22	2013-04-28	2012-10-06	2013-04-28		
DM	CDISC010	1236	2013-09-21	2013-09-26	2013-09-21	2013-09-21	2013-09-08	2013-09-26		
DM	CDISC011	1336	2012-12-07	2013-06-05	2012-12-07	2013-06-05	2012-11-21	2013-07-05		
DM	CDISC012	1378	2013-09-03	2014-01-28	2013-09-03	2014-01-28	2013-08-24	2014-01-28		
DM	CDISC013	1083	2013-07-22	2013-08-03	2013-07-22	2013-08-01	2013-07-09	2013-08-03	2013-08-03	Y
DM	CDISC014	1012	2013-04-03	2013-05-02	2013-04-03	2013-04-29	2013-03-20	2013-09-18		
DM	CDISC015	1022					2014-03-17	2014-03-17		
DM	CDISC016	1143	2013-04-03	2013-06-01	2013-04-03	2013-05-30	2013-03-30	2013-09-22		
DM	CDISC017	1250	2013-09-21	2014-02-08	2013-09-21	2014-01-31	2013-08-21	2014-03-08		



141	CORE-000238	EX.EXENDTC	fully executable	dm.json	CDISC008	8	\$max_ex_exstatc, RFXENDTC	2014-10-31, 2014-10-31, 2014-10-31
142	CORE-000239	RFXSTDTC does not equal the earliest value of EX.EXSTDTC	fully executable	dm.json	CDISC001	3	\$min_ex_exstdtc, RFXSTDTC	2012-11-30, 2013-08-29



Developing custom rules



Developing custom rules

- CORE not only enables the development of "custom" rules, but also allows these rules to be added to the CORE engine as a "custom standard"
- Custom rules can be generated in either YAML or JSON
- Custom rules can be grouped into "custom standards", e.g. per study or therapeutic area (sponsors), per sponsor (CROs), ... and versioned
- Custom rules can however also be executed individually

Developing custom rules

- For editing in YAML or JSON, one can use one of the many available editors (also online).
- We are also developing an (open source) YAML editor implementing the CORE schema

```
Check:
  all:
    # Raise an issue when AGE (in years) is higher than 65
    - name: AGEU
      operator: equal_to
      value: YEARS
    - name: AGE
      operator: greater_than
      value: 65

Core:
  Id: XML4Pharma_AGE_65
  Status: Draft
  Version: '1'
  Description: 'Age in years must be 65 or lower'
  Executability: Fully Executable

Outcome:
  Message: 'Age in years must be 65 or lower'
  Rule Type: Record Data

Scope:
  Classes:
    Include:
      - SPECIAL PURPOSE
  Domains:
    Include:
      - DM
  Sensitivity: Record
```

Developing custom rules - Examples

```
[-] Check:
[-] all:
    # Raise an issue
    # is not one of
    # REMARK that VSI
    - name: VSTESTCD
      operator: is_co
      value:
        - DIABP
        - SYSBP
    - name: VSPOS
      operator: exist
    - name: VSPOS
      operator: non_e
    - name: VSPOS
      operator: is_no
      value:
        - PRONE
        - SEMI-RECUMI
        - SITTING
        - STANDING
        - SUPINE
[-] Core:
    Id: XML4Pharma_VSP
    Status: Draft
    Version: '1'
    Description: "VSPOS
    Executability: Fully
[-] Outcome:
    Message: For VSTES
    Rule Type: Record Da
[-] Scope:
    Classes:
      Include:
        - FINDINGS
    Domains:
      Include:
        - VS
    Sensitivity: Record
```

```
[-] Check:
[-] all:
    # Preconditions (selections)
    - name: 'LBTESTCD'
      operator: equal_to
      value: 'WBC'
    # LBORRES is populated
    - name: 'LBORRES'
      operator: non_empty
    # Postcondition (under which a violation is generated)
    # Data type must be float
    - name: 'LBORRES'
      operator: not_matches_regex
      value: '^\\d*\\.?.?\\d*$'
```

```
[-] Outcome:
    Message: 'When LBORRES is populated, its value for LBTESTCD=WBC must be a floating point number'
    Rule Type: Record Data
[-] Scope:
    Classes:
      Include:
        - FINDINGS
    Domains:
      Include:
        - LB
    Sensitivity: Record
```

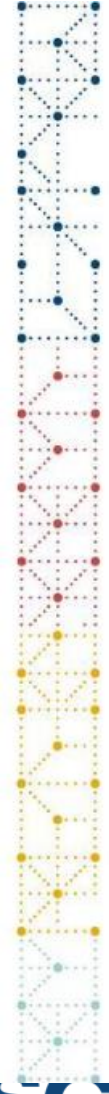
Some rules already look related to "SDTM Dataset Specializations", which are implementations of CDISC Biomedical Concepts



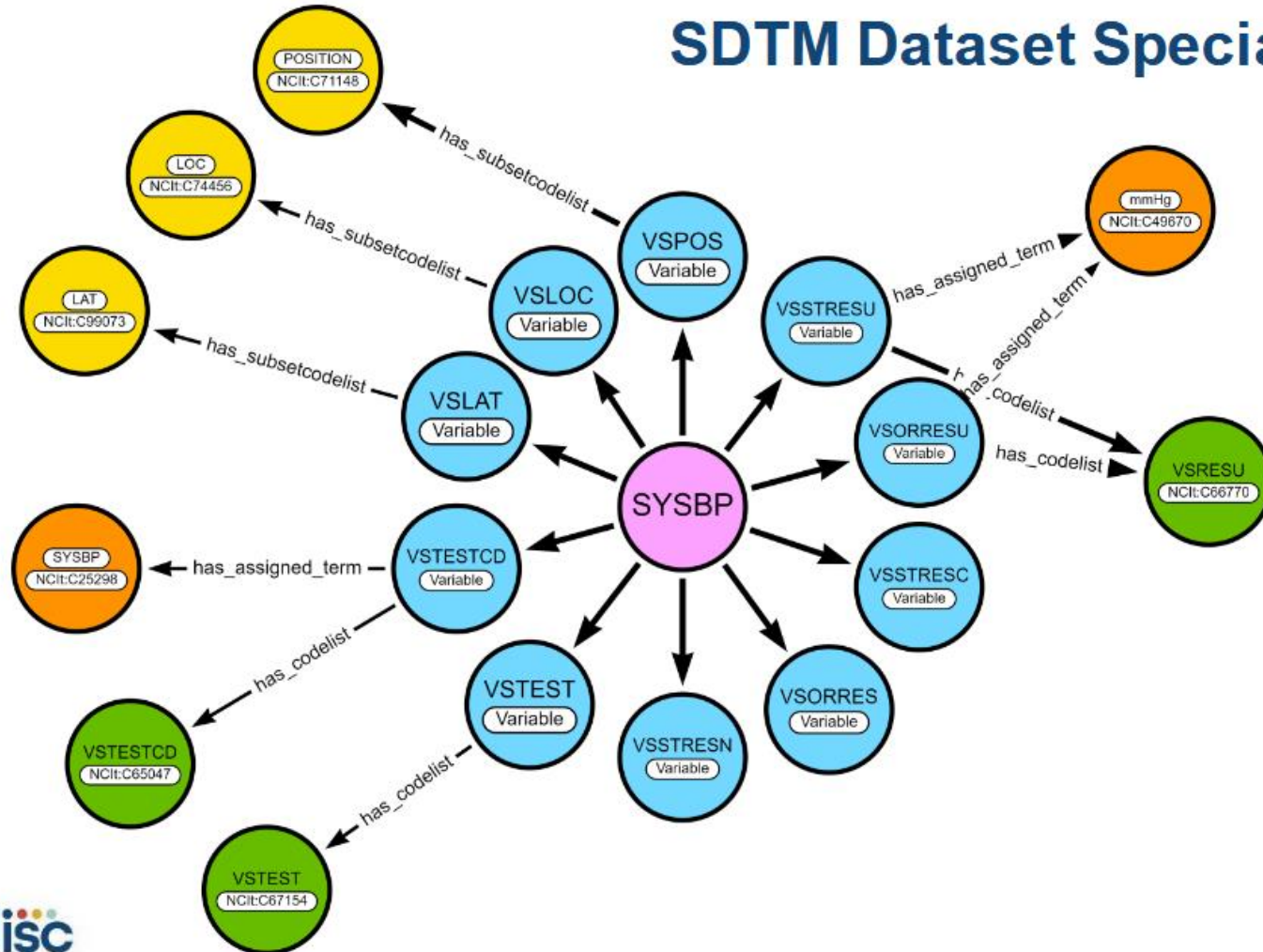
Developing CORE Rules from SDTM Dataset Specializations



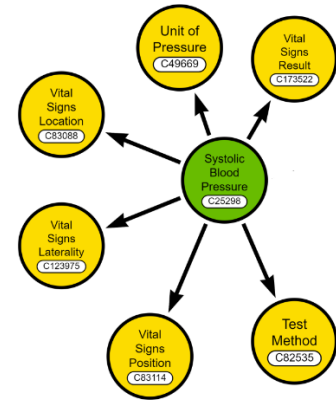
SDTM Dataset Specializations as a source for "custom" CORE rules



SDTM Dataset Specializations



April 2026:
1326 Dataset Specializations
in the CDISC Library



BC



As these "SDTM Dataset Specializations" essentially define "rules" (or at least "best practices"), can't we generate CORE rules in YAML format from them in an automated way ?

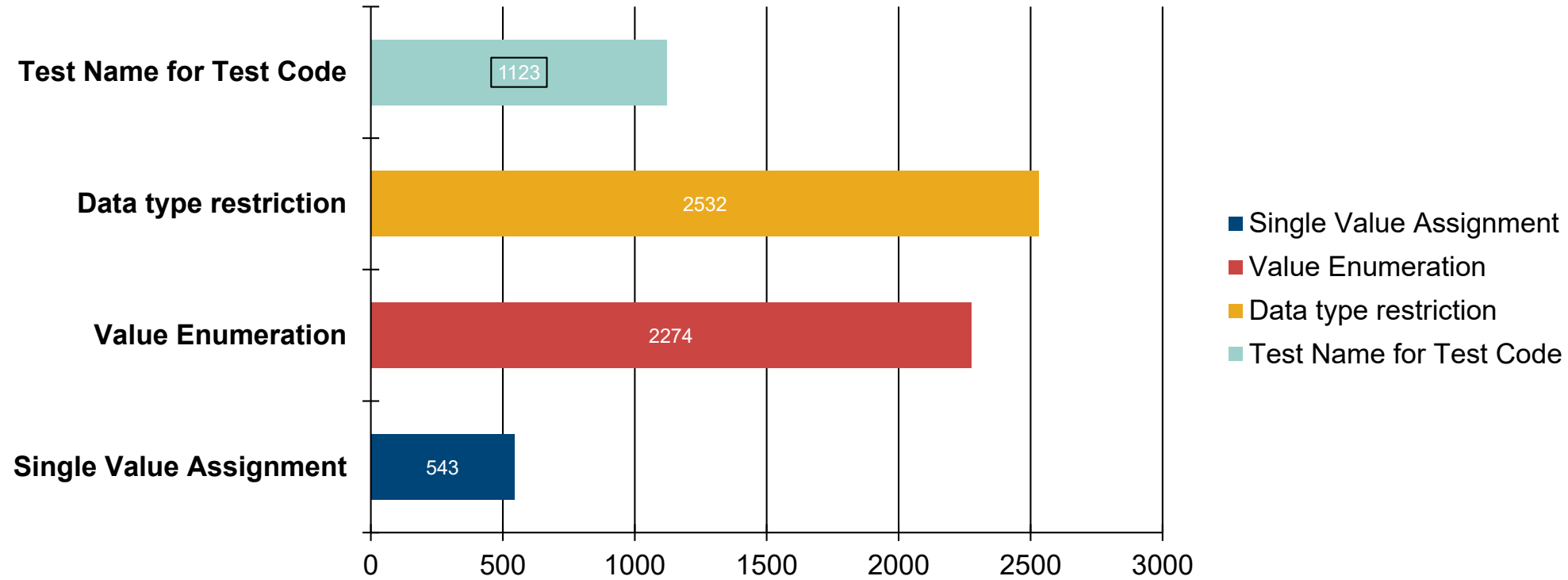


Process / Algorithm

- Query the CDISC Library using the API for all "Dataset Specializations"
- Get the JSON representation for each of them
- Look for the objects without "vlmTarget" attribute
=> **Precondition** (e.g. VSTESTCD=WSTCIR)
- Look for the objects for which vlmTarget=true
=> **Postcondition** (e.g. VSORRES=float, VSORRESU=cm,in,mm)
- For **each postcondition**, generate the YAML code for a CORE rule
- Save the YAML to file

Result: 6472 CORE rules (one YAML file per rule)

Rule types



Example WSTCIR (Waist Circumference)

(SDTM Dataset Specialization JSON)

```
{
  "isNonStandard": false,
  "comparator": "EQ",
  "role": "Topic",
  "codelist": {
    "submissionValue": "VSTESTCD",
    "conceptId": "C66741",
    "href": "https://evsexplore.semantics.cancer.gov/evsexplore/concept/ncit/C66741"
  },
  "name": "VSTESTCD",
  "mandatoryVariable": true,
  "mandatoryValue": true,
  "relationship": {
    "linkingPhrase": "is the code for the value in",
    "subject": "VSTESTCD",
    "predicateTerm": "IS_DECODED_BY",
    "object": "VSTEST"
  },
  "assignedTerm": {
    "conceptId": "C100948",
    "value": "WSTCIR"
  }
}
```

Precondition ("where")

WSTCIR =
Waist Circumference

Example WSTCIR (Waist Circumference)

(SDTM Dataset Specialization JSON)

```
{
  "name": "VSORRES",
  "dataElementConceptId": "C70856",
  "isNonStandard": false,
  "role": "Qualifier",
  "dataType": "float",
  "length": 8,
  "format": "8.3",
  "significantDigits": 3,
  "relationship": {
    "subject": "VSORRES",
    "linkingPhrase": "is the res",
    "predicateTerm": "IS_RESULT_",
    "object": "VSTESTCD"
  },
  "mandatoryVariable": true,
  "mandatoryValue": false,
  "originType": "Collected",
  "originSource": "Investigator",
  "vlmTarget": true
},
```

```
{
  "name": "VSORRESU",
  "dataElementConceptId": "C42578",
  "isNonStandard": false,
  "codelist": {
    "conceptId": "C66770",
    "submissionValue": "VSRESU",
    "href": "https://evsexplore.semantics.cancer.gov/evsexplore/concept/ncit/C66770"
  },
  "subsetCodelist": "VSRESU_WSTCIR",
  "valueList": [
    "cm",
    "in",
    "mm"
  ],
  "role": "Qualifier",
  "relationship": {
    "subject": "VSORRESU",
    "linkingPhrase": "is the unit for the value in",
    "predicateTerm": "IS_UNIT_FOR",
    "object": "VSORRES"
  },
  "mandatoryVariable": true,
  "mandatoryValue": false,
  "vlmTarget": true
},
```

Result / Postcondition

WSTCIR =

Waist Circumference

Example WSTCIR (Waist Circumference)

(CORE rule in YAML)

```
# CORE rule code for Dataset Specialization with Id: WSTCIR
```

```
- Check:
```

```
  all:
```

```
    # Preconditions (selections)
```

```
    # variable value of 'VSTESTCD' must be equal to 'WSTCIR'
```

```
    - name: 'VSTESTCD'
```

```
      operator: equal_to
```

```
      value: 'WSTCIR'
```

```
    # Populated-condition (when variable 'expected' or 'permissible')
```

```
    - name: 'VSORRES'
```

```
      operator: non_empty
```

```
    # Postcondition (under which a violation is generated)
```

```
    # Data type must be float
```

```
    - name: 'VSORRES'
```

```
      operator: not_matches
```

```
      value: '^\d*\.\?\d*$'
```

```
- Core:
```

```
  Id: DSS_VS_WSTCIR_1
```

```
  Status: Draft
```

```
  Version: '1'
```

```
# CORE rule code for Dataset Specialization with Id: WSTCIR
```

```
- Check:
```

```
  all:
```

```
    # Preconditions (selections)
```

```
    # variable value of 'VSTESTCD' must be equal to 'WSTCIR'
```

```
    - name: 'VSTESTCD'
```

```
      operator: equal_to
```

```
      value: 'WSTCIR'
```

```
    # Populated-condition (when variable 'expected' or 'permissible')
```

```
    - name: 'VSORRESU'
```

```
      operator: non_empty
```

```
    # Postcondition (under which a violation is generated)
```

```
    - name: 'VSORRESU'
```

```
      operator: is_not_contained_by
```

```
      value:
```

```
        - 'cm'
```

```
        - 'in'
```

```
        - 'mm'
```

```
- Core:
```

```
  Id: DSS_VS_WSTCIR_2
```

```
  Status: Draft
```

```
  Version: '1'
```



**6472 automatically generated
custom CORE rules is a lot ...**

Can we trust them?

And this number will still grow
when new BCs and Dataset Specializations are
published by CDISC



Automation and Curation

- Curation may take a lot of time ... but ...
- Most rules are very simple:
 - Test name for test code
 - data type of --ORRES, --STRESC
 - enumeration of allowed values for --ORRESU, --STRESU
 - enumeration of allowed values for --SPEC, --LOC, ...
- but even then ...

Example WBC - LBSPEC=BLOOD

```
# CORE rule code for Dataset Specialization with Id: WBCBLD
Check:
  all:
    # Preconditions (selections)
    # variable value of 'LBTESTCD' must be equal to 'WBC'
    - name: 'LBTESTCD'
      operator: equal_to
      value: 'WBC'
    - any:
      # Uncomment the following lines when the rule
      # is also applicable when the variable is not populated
      # # Variable is not mandatory to be populated
      # - name: 'LBSPEC'
      #   operator: empty
      # variable value of 'LBSPEC' must be equal to 'BLOOD'
      - name: 'LBSPEC'
        operator: equal_to
        value: 'BLOOD'
    # Populated-condition (when variable 'expected' or 'permissib:
    - name: 'LBSTRESU'
      operator: non_empty
    # Postcondition (under which a violation is generated)
    - name: 'LBSTRESU'
      operator: is_not_contained_by
      value:
        - '10^9/L'
```

- LBSPEC is "permissible" !
- Should the rule still apply when LBSPEC was not populated?
- For LBSPEC=URINE, expected LBSTRESU='10^6/L'!

Software implementation

- We added the 6472 rules as a "custom standard" named "**cdisc_dataset_specializations**", currently as version 1.3, to the CORE engine [*]
- and deployed that to several of our software tools:
 - Open source "Smart Submission Dataset Viewer"
 - "SDTM-ETL" mapping software

[*]: For "how to", see: <https://github.com/cdisc-org/cdisc-rules-engine/>

Software implementation - Graphical User Interface

☐ Perform post-processing for assigning --LOBXFL

☒ Split records > 200 characters to SUPP-- records

☒ Move non-standard SDTM Variables to SUPP--

☐ Move Relrec Variables to Related Records (RELREC) domain

☒ View Result SDTM tables

☐ Generate 'NOT DONE' records for QS datasets

☐ Unique --SEQ values across 'split' domains

☒ Save Result SDTM tables as:

☐ Perform post-processing for assigning --LOBXFL

☐ Move Comment Variables to SUPP--

☐ Try to generate 1:N Relationships

☒ Adapt Variable Lengths

☐ Re-sort records using

☒ Perform CDISC CORE

☐ Dataset-JSON 1.1

☒ SAS-XPT

☐ UTF-8 encoded CSV

☐ SQL

SDTM export files directory:

☒ SAS-XPT

☐ UTF-8 encoded CSV

☐ SQL INSERT statements

define.xml

for 'split' domain d

LOINC code 704-7

value for LOINC code

.OOD

LOINC code 704-7

Message

Folder with SAS-XPT files: D:\temp

☒ dm.xpt

☒ lb.xpt

☒ vs.xpt

Select All

Select None

SAS-XPT files to be validated:

☒ DSS_IS_ARIGAABMMA_3

☒ DSS_IS_ARIGAABMMA_4

☒ DSS_IS_ARIGAABMMA_5

☒ DSS_IS_ARIGAABMUCOR_1

Authority: XML4Pharma

Trigger violation when ISTECD='ARIGAAB' AND ISBDAGNT='MOLD MIX ANTIGENS' AND ISSPEC is one of: 'SERUM','SERUM

is NOT 'umol/L'

Applicable standards and versions:

☒ SDTMIG 3.4

☒ DSS_IS_ARIGAABOGP_1

☒ DSS_IS_ARIGAABOGP_2

☒ DSS_IS_ARIGAABOGP_3

☒ DSS_IS_ARIGAABOGP_4

Select All

Select None

Rules used for Validation:

☒ DSS_IS_ARIGAABOGP_1

☒ DSS_IS_ARIGAABOGP_2

☒ DSS_IS_ARIGAABOGP_3

☒ DSS_IS_ARIGAABOGP_4

Select All

Select None

Explore Validation Rules

Report format:

☒ Excel

☐ JSON

Report Folder: D:\temp

Browse...

Execute CORE Validation

CORE rulesets

?

☐ CDISC/FDA Ruleset

☒ CDISC Dataset Specializations Ruleset (custom rules)

☐ Own custom data quality rule(s)

OK

cdisc

Software implementation - Selections

Core Rules Exploration

LB
MB
MI

Filter

Reset

Select all rules shown
Clear selection

☐ Rule ID
☒ Domains Included
☐ Sensitivity
☐ Domains Excluded
☐ Authorities
☐ Rule Description
☐ Standards and Versions
☐ Message
☐ Classes Included
☐ References
☐ Classes Excluded

Include	Rule ID	Authority	Sensitivity	Standards and Versio...	Classes Included	Classes Exclud...	Domains Includ...	Domains Exclu...	Description	Message	References
☐	DSS_LB_BILDIRSERPL_5	XML4Pharma	Record	SDTMIG 3.2 SDTMIG 3.3 SDTMIG 3.4	ALL		LB		Trigger violation when LBTESTCD='BILD...	For LBTESTCD='BILDIR'; LBS...	
☐	DSS_LB_BILISERPL_1	XML4Pharma	Record	SDTMIG 3.2 SDTMIG 3.3 SDTMIG 3.4	ALL		LB		Trigger violation when LBTESTCD='BIL' ...	For LBTESTCD='BILI', LBORR...	
☐	DSS_LB_BILISERPL_2	XML4Pharma	Record	SDTMIG 3.2 SDTMIG 3.3 SDTMIG 3.4	ALL		LB		Trigger violation when LBTESTCD='BIL' AND LBSPEC is one of: 'SERUM OR PLASMA'; 'SERUM'; 'PI...	For LBTESTCD='BILI', LBORRESU is populated AND LBORRESU is NOT one of: 'mg/dL'; 'mg/L'; 'mmol/L'; 'umol/L'	
☐	DSS_LB_BILISERPL_3	XML4Pharma	Record	SDTMIG 3.2 SDTMIG 3.3 SDTMIG 3.4	ALL		LB		Trigger violation when LBTESTCD='BIL' ...	For LBTESTCD='BILI', LBSTR...	
☐	DSS_LB_BILISERPL_4	XML4Pharma	Record	SDTMIG 3.2 SDTMIG 3.3 SDTMIG 3.4	ALL		LB		Trigger violation when LBTESTCD='BIL' ...	For LBTESTCD='BILI', LBSTR...	
☐	DSS_LB_BILISERPL_5	XML4Pharma	Record	SDTMIG 3.2 SDTMIG 3.3 SDTMIG 3.4	ALL		LB		Trigger violation when LBTESTCD='BIL' ...	For LBTESTCD='BILI', LBSTR...	
☐	DSS_LB_BILIURINPRES_1	XML4Pharma	Record	SDTMIG 3.2 SDTMIG 3.3 SDTMIG 3.4	ALL		LB		Trigger violation when LBTESTCD='BIL' ...	For LBTESTCD='BILI', LBORR...	
☐	DSS_LB_BILIURINPRES_2	XML4Pharma	Record	SDTMIG 3.2 SDTMIG 3.3 SDTMIG 3.4	ALL		LB		Trigger violation when LBTESTCD='BIL' ...	For LBTESTCD='BILI', LBSTR...	
☐	DSS_LB_BILIURIN_1	XML4Pharma	Record	SDTMIG 3.2 SDTMIG 3.3 SDTMIG 3.4	ALL		LB		Trigger violation when LBTESTCD='BIL' ...	For LBTESTCD='BILI', LBORR...	
☐	DSS_LB_BILIURIN_2	XML4Pharma	Record	SDTMIG 3.2 SDTMIG 3.3 SDTMIG 3.4	ALL		LB		Trigger violation when LBTESTCD='BIL' ...	For LBTESTCD='BILI', LBORR...	

OK

Software implementation - GUI view

Export selected table as SAS-XPT		Export selected table as Dataset-JSON 1.1		Export selected table as UTF-8 encoded		
Perform CORE validation on selected table						
CES:LB	CES:VS	CES:SUPPDM	CES:SUPPVS			
LB.LBTEST	LB.LBCAT	LB.LBORRES	LB.LBORRESU	LB.LBORNRL0	LB.LBORNRLHI	LB.L
Erythrocytes	HEMATOLOGY	4.9	10^9/L	4.7	6.1	4.9
Leukocytes	HEMATOLOGY	6.2	10^9/L	4.3	10.8	6.2
Erythrocytes	HEMATOLOGY	5.1	10^9/L	4.7	6.1	5.1
Leukocytes	HEMATOLOGY	6.4	10^9/L	4.3	10.8	6.4
Erythrocytes	HEMATOLOGY	5.1	10^9/L	4.7	6.1	5.1
Leukocytes	HEMATOLOGY	6.5	10^9/L	4.2	10.7	6.5
Erythrocytes	HEMATOLOGY	5.4	10^9/L	4.7	6.1	5.4
Leukocytes	HEMATOLOGY	6.6	10^9/L	4.3	10.8	6.6
Erythrocytes	HEMATOLOGY	5.0	10^9/L	4.7	6.1	5.0
Leukocytes	HEMATOLOGY	5.9	10^9/L	4.3	10.8	5.9
Erythrocytes	HEMATOLOGY	5.2	10^9/L	4.7	6.1	5.2
Leukocytes	HEMATOLOGY	6.3	10^9/L	4.3	10.8	6.3
Erythrocytes	HEMATOLOGY	5.1	10^9/L	4.7	6.1	5.1
Leukocytes	HEMATOLOGY	6.6	10^9/L	4.2	10.7	6.6
Erythrocytes	HEMATOLOGY	5.5	10^9/L	4.7	6.1	5.5
Leukocytes	HEMATOLOGY	6.1	10^9/L	4.3	10.8	6.1
Erythrocytes	HEMATOLOGY	4.9	10^9/L	4.7	6.1	4.9
Leukocytes	HEMATOLOGY	6.0	10^9/L	4.3	10.8	6.0
Erythrocytes	HEMATOLOGY	5.2	10^9/L	4.7	6.1	5.2
Leukocytes	HEMATOLOGY	6.7	10^9/L	4.3	10.8	6.7

For LBTESTCD='RBC', LBORRESU is expected to be: '10^12/L'

Summary

- For CORE, custom ("data quality") rules can easily be developed
- Groups of "custom rules" can be added to the CORE engine as "custom standards" and versioned
- We generated a large number of "custom rules" in an automated way starting from the "SDTM Dataset Specializations", and added these to the CORE engine as a "custom standard", and deployed them into several of our software tools
- Graphical User Interfaces (GUIs) can easily be build "on top" of CORE to make the use of CORE extremely user-friendly

Thank You!

Grazie mille per l'attenzione!

