CORE: Checking USDM for Conformance

25 June 2024

Agenda

- Digital Data Flow and USDM
- Conformance Rule Coverage for USDM
- CORE Demo
- Planning for phase 4

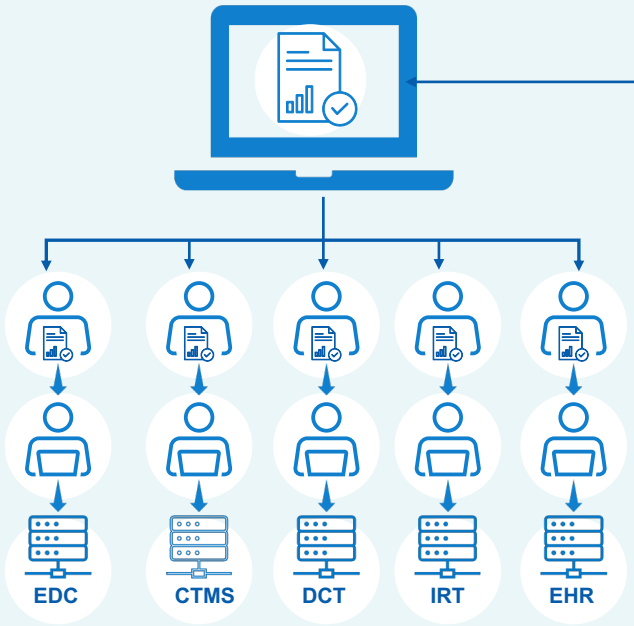
Digital Data Flow and USDM

https://www.transceleratebiopharmainc.com/assets/digital-data-flow-solutions/

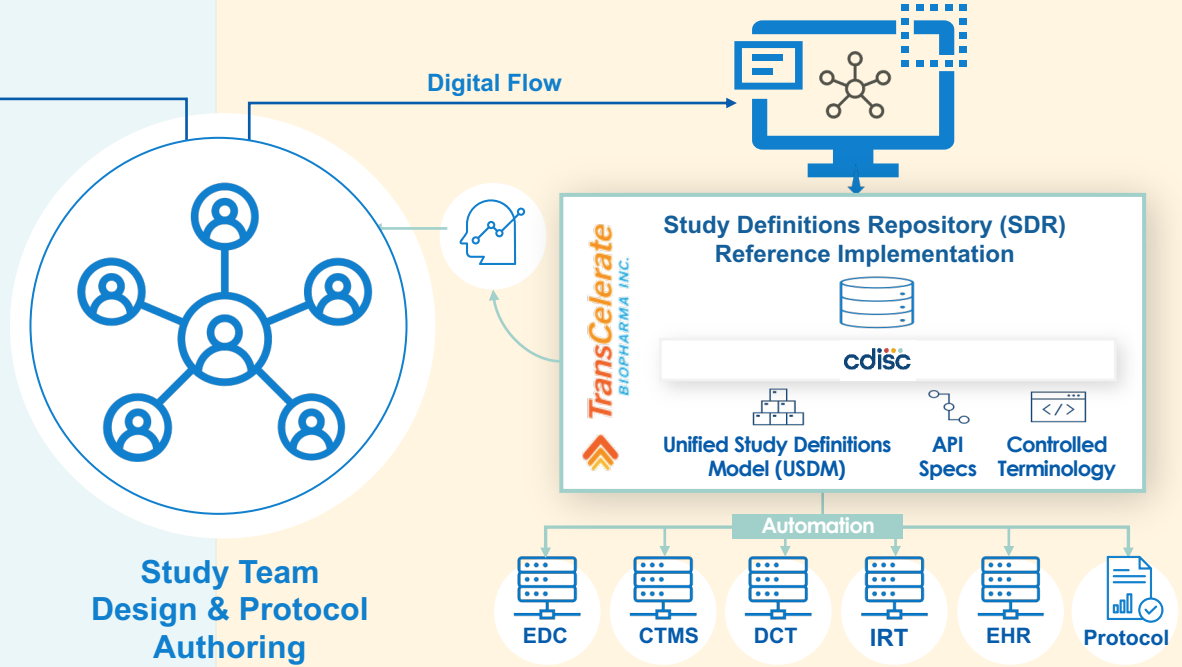
TransCelerate Digital Data Flow (DDF) Ambition

Write Once, Read Many

TODAY: Document-based paradigm for protocol creation, interpretation, and transcription into consuming systems



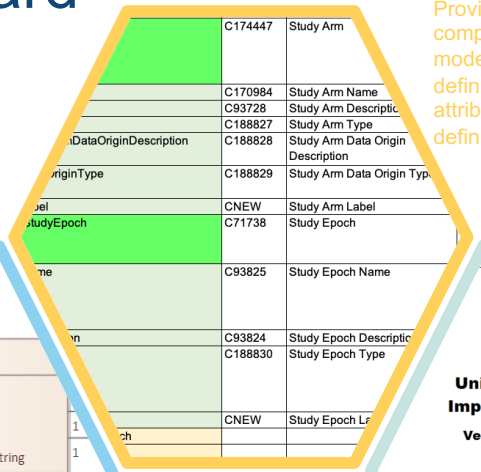
TOMORROW: Digital paradigm for protocol creation, with fully automated data flow and interoperability between systems



The USDM Standard

CDISC Controlled Terminology

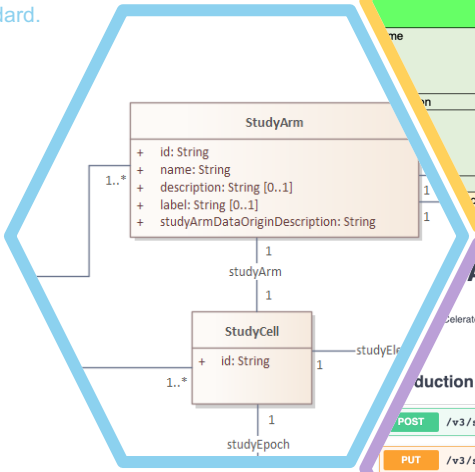
Provides further semantics, complementing the UML model. Includes the definition of classes and attributes along with the definition of value sets



	C174447	Study Arm
	C170984	Study Arm Name
	C93728	Study Arm Descriptio
	C188827	Study Arm Type
DataOriginDescription	C188828	Study Arm Data Origin Description
riginType	C188829	Study Arm Data Origin Typ
el	CNEW	Study Arm Label
tudyEpoch	C71738	Study Epoch
me	C93825	Study Epoch Name
n	C93824	Study Epoch Descriptio
	C188830	Study Epoch Type
	CNEW	Study Epoch La
ch		

Logical Model

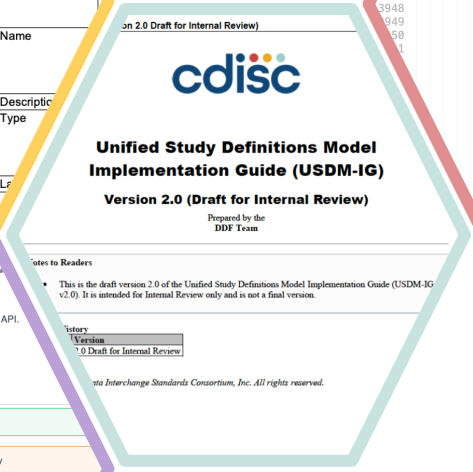
The UML logical model (a class diagram) that provides the basis for the USDM standard.



Examples

Example protocols implemented in the USDM with associated JSON files and visualisations

```
"studyArms": [
  {
    "id": "StudyArm_1",
    "name": "Placebo",
    "label": "",
    "description": "Placebo",
    "type": {
      "id": "Code_61",
      "code": "C174268",
      "codeSystem": "http://www.cdisc.org",
      "codeSystemVersion": "2022-12-16",
      "decode": "Placebo Comparator Arm"
    },
    "studyArmDataOriginDescription": "Data collec
    "dataOriginType": {
      "id": "Code_62",
      "code": "C188866",
      "codeSystem": "http://www.cdisc.org",
      "codeSystemVersion": "2022-12-16",
      "decode": "Data Generated Within Study"
    }
  },
  {
    "id": "StudyArm_2",
    "name": "Xanomeline Low Dose",
    "label": "",
    "description": "Active Substance",
    "type": {
      "id": "Code_63",
      "code": "C174267",
      "codeSystem": "http://www.cdisc
      "codeSystemVersion": "2022-12-
```



on 2.0 Draft for Internal Review)

cdisc

Unified Study Definitions Model Implementation Guide (USDM-IG)

Version 2.0 (Draft for Internal Review)

Prepared by the DDF Team

otes to Readers

- This is the draft version 2.0 of the Unified Study Definitions Model Implementation Guide (USDM-IG v2.0). It is intended for Internal Review only and is not a final version.

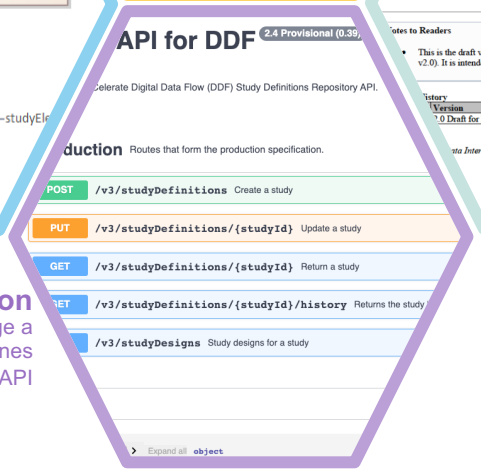
Version

.0 Draft for Internal Review

nta Interchange Standards Consortium, Inc. All rights reserved.

API Specification

Provides the means to exchange a single study between machines using a JSON API



API for DDF 2.4 Provisional (0.39)

celerate Digital Data Flow (DDF) Study Definitions Repository API.

duction Routes that form the production specification.

POST /v3/studyDefinitions Create a study

PUT /v3/studyDefinitions/{studyId} Update a study

GET /v3/studyDefinitions/{studyId} Return a study

ET /v3/studyDefinitions/{studyId}/history Returns the study

/v3/studyDesigns Study designs for a study

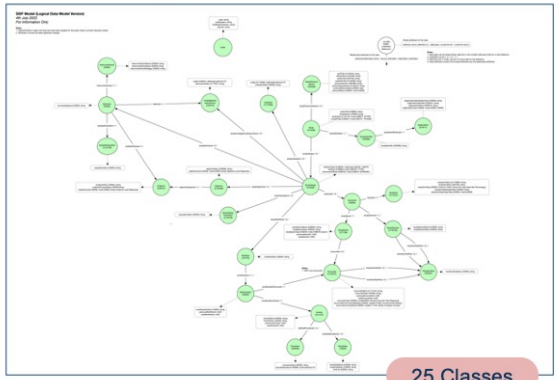
Expand all object

Implementation Guide

Guidance on using the USDM model and ensuring conformance with the standard

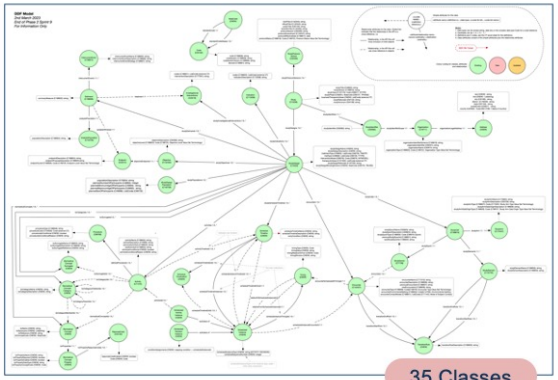
CDISC DDF / USDM: Phases One, Two and Three

Phase One



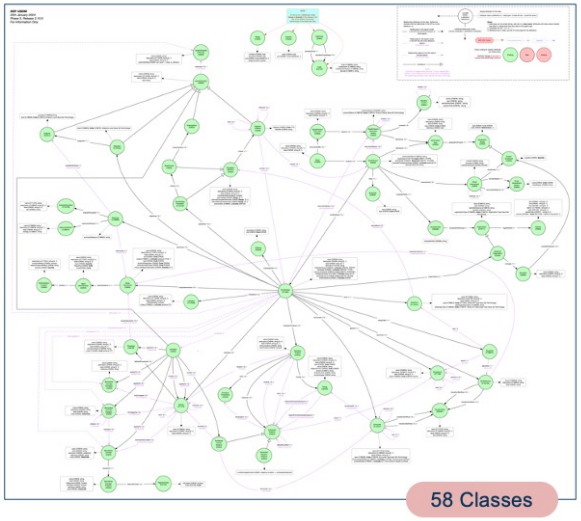
- Solid foundation
- The protocol document was an external entity into which the structured content could be exported

Phase Two



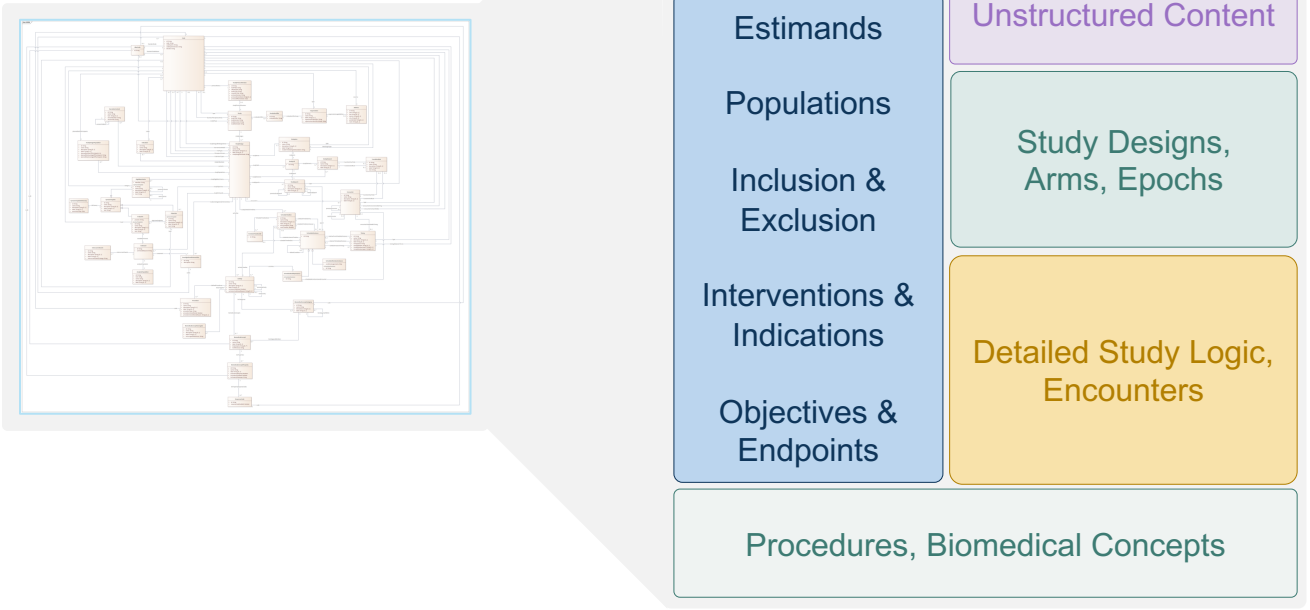
- Focused on the structured elements of the protocol, e.g. the Schedule of Activities (SoA) & Biomedical Concepts (BCs)
- The protocol document still an external entity

Phase Three



- Now contains structured and unstructured elements
- The entire protocol document can be held within the USDM
- Allows for the protocol document to be generated from the model

USDM Content



- **Conformance Rule Coverage for USDM**

Latest page content update: 17 Jan 2024

Introduction

CDISC Conformance Rules are an integral part of the **Foundational Standards** and serve as the specific guidance to Industry for the correct implementation of the Standards in clinical studies. An emerging Industry best practice is to use Conformance Rules on an ongoing basis, throughout the study, to keep the data as close to submission ready as possible and to ensure quality in all data exchange scenarios.

Current CDISC Conformance Rules need to be expressed in a common specification to be loaded to the **CDISC Library**. In addition, an executable component must be developed for every Conformance Rule.

Project Goals and Objectives

The overall goal of the CORE Project is to deliver a governed set of unambiguous and executable Conformance Rules for each Foundational Standard, and to provide a Reference Implementation of an open-source execution engine for the executable Rules.

The global clinical research community will be able to leverage the free and open CORE software to test study data for conformance to CDISC standards as well as to regulatory and sponsor-specific conformance rule sets.

The CORE Project objectives are to:

- Ensure each standard has a set of unambiguous, executable Conformance Rules
- Ensure consistency across Conformance Rule implementations
- Expedite the availability of executable Conformance Rules for new Foundational Standards
- Create executable Conformance Rules vetted by the CDISC standards development teams
- Create a Reference Implementation of an open-source engine that executes the Rules
- Release the open-source engine under the **CDISC Open-Source Alliance (COSA)**

https://www.cdisc.org/core

USDM CORE rules

- Purpose
  - Check that exchanged JSON API file is correct according to USDM logic and implementation guide
  - Inform users of incorrect and/or unlikely content via pre-specified rules
  - Inform users of correct implementation via logical representation of content rules
- Phase 3
  - Proof Of Concept
- Phase 4
  - Rule set aligning with USDM 3.0
  - Rule set aligning with USDM 4.0

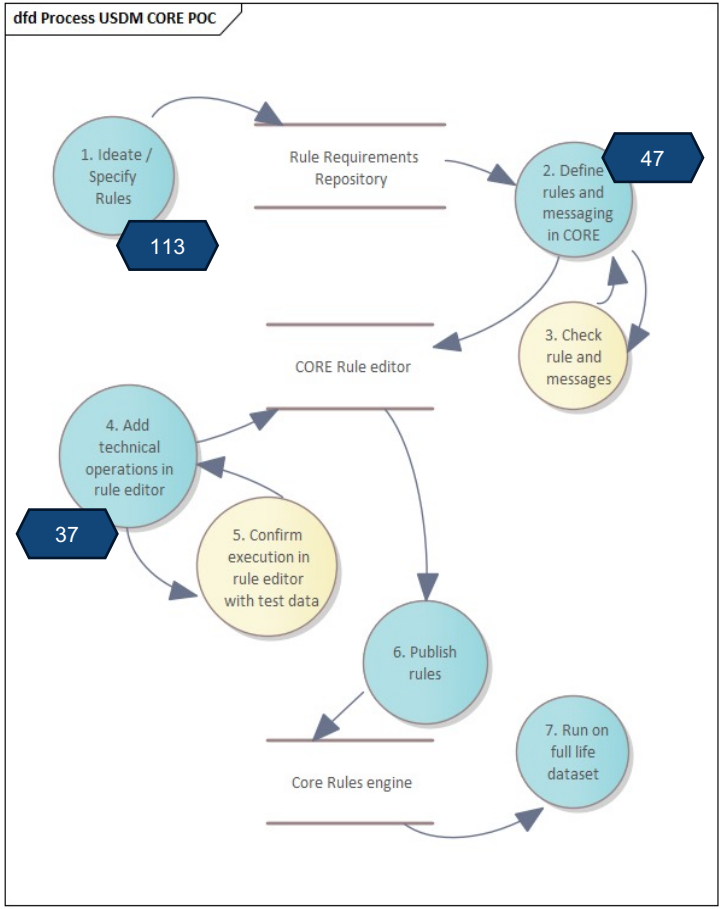
cdisc

DDF3A CORE POC Use Case and Scope

- Demonstrating that USDM JSON files are USDM compliant,
  - e.g., Transfer of USDM JSON file from one organization to another (e.g., Vendor to Sponsor)
- Scope
  - Develop a representative set of conformance rule specifications (up to 100)
  - In addition, develop a representative set of machine executable Conformance rules (in YAML) to cover a wide breadth of the different types of rules to demonstrate DDF conformance
- Details
  - Enhance the CORE Rule Editor and schema to handle various DDF rule types
  - Develop and test a representative set of machine executable conformance rules (in YAML format)
  - Enhance the CORE open-source engine to run these Conformance Rules against a USDM JSON file
  - Scope CDISC Library modifications (to store DDF conformance rules and the USDM model)

cdisc

Process

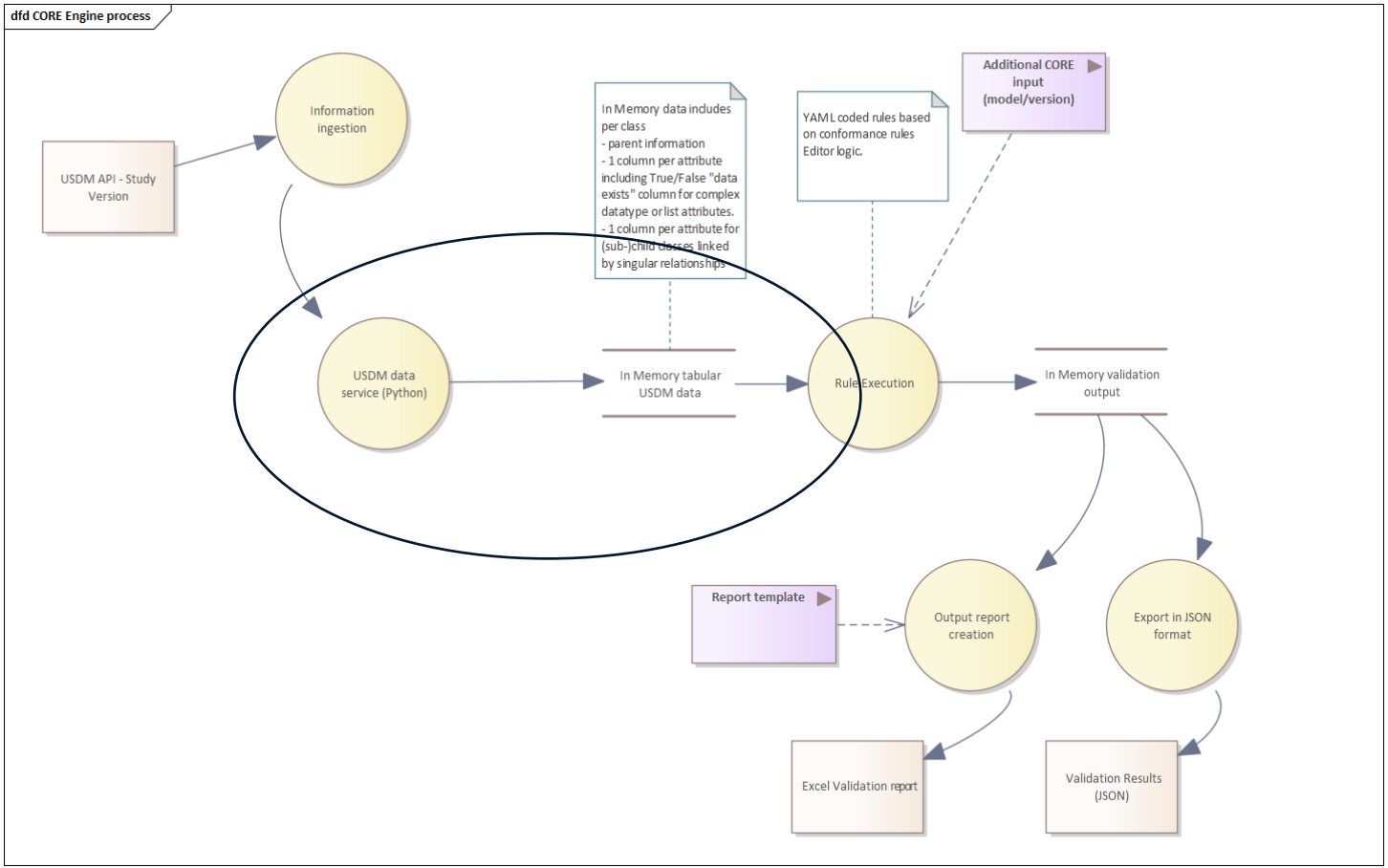

dfd Process USDM CORE POC
1. Ideate / Specify Rules
113
Rule Requirements Repository
2. Define rules and messaging in CORE
47
3. Check rule and messages
CORE Rule editor
4. Add technical operations in rule editor
37
5. Confirm execution in rule editor with test data
6. Publish rules
7. Run on full life dataset
Core Rules engine

Test Data Template

- Excel workbook based on existing SDTM test data template
  - Designed to align with current Rules Editor functionality (e.g., .xpt suffix for sheet names) to minimize effort required for POC
  - Link between json and Core engine
- Programmatically generated from USDM UML, CT and API
  - In general, 1 sheet per USDM class/entity
  - Linking attributes retained

parent_entity	parent_id	parent_rel	id	standardCode.id	standardCode.code	standardCode.codeSystem	standardCode.codeSystemVersion	standardCode.decode
Parent Entity Name	*Parent Entity Id*	*Name of Relationship from Parent Entity*	*(Alias Code Id)*	*(Alias Code Standard Code) / (Code Id)*	*(Alias Code Standard Code) / Code Value*	*(Alias Code Standard Code) / Code System Name*	*(Alias Code Standard Code) / Code System Version*	*(Alias Code Standard Code) / Decode*
String	*String*	*String*	*String*	*String*	*String*	*String*	*String*	*String*
[1]	*[1]*	*[1]*	*[1]*	*Code[1].id[1]*	*Code[1].code[1]*	*Code[1].codeSystem[1]*	*Code[1].codeSystemVersion[1]*	*Code[1].decode[1]*

Core engine process updates in POC



Running the CORE Engine

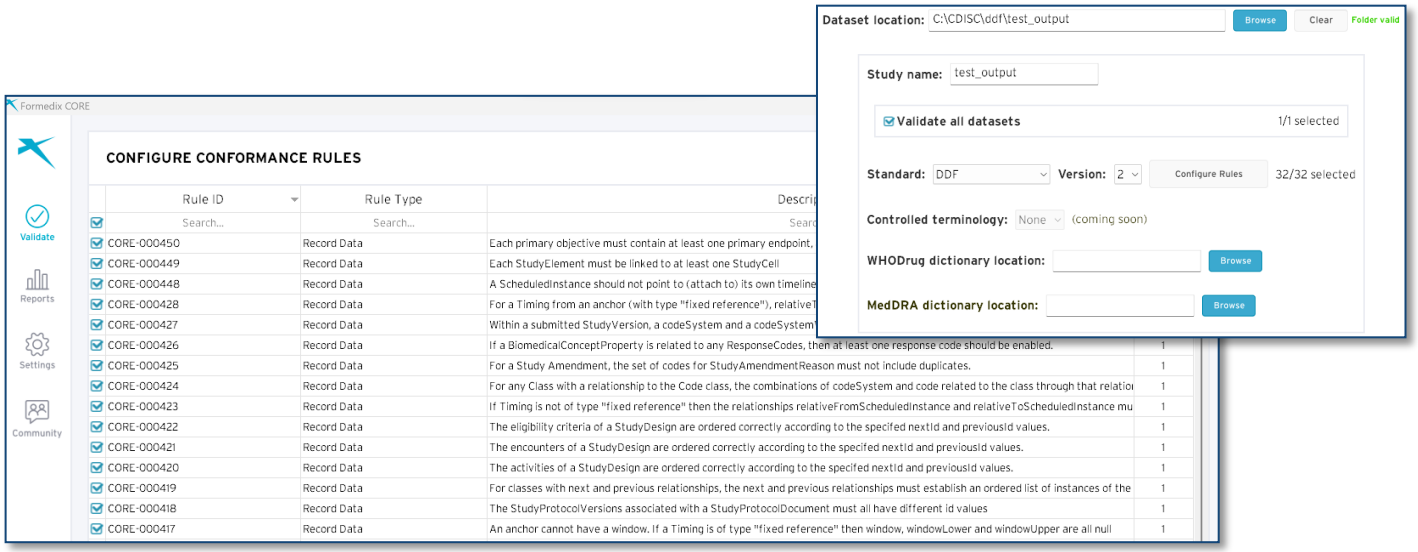
- CLI executable available in GitHub
  - Cached rules
  - Windows, Mac, and Linux install packages
  - Unzip and run
  - Will need datasets to validate

```
(venv) PS C:\CDISC\cdisc-rules-engine> python core.py validate -s ddf -v 2 -dp ..\ddf\test_output\CDISC_Pilot_Study_dirty1.json
C:\CDISC\cdisc-rules-engine
  [                                ]  100%
(venv) PS C:\CDISC\cdisc-rules-engine>
```

- Engine available on PyPI
  - Engine is a component that can be used in your own code

Running the CORE Engine

- Desktop versions
  - Vendor released versions of CORE
  - Includes a user-friendly UI
  - Easier for non-technical users to evaluate



Conversion of USDM JSON File

- The CORE engine converts each USDM class in the selected JSON file into tabular format before applying the rules.

```
{
  "study": {
    "id": null,
    "name": "Study_CDISC PILOT - LZZT",
    "description": null,
    "label": null,
    "versions": [
      {
        "id": "StudyVersion_1",
        "studyTitle": "Safety and Efficacy of the Xanomeline Transdermal Therapeutic System
        "versionIdentifier": "2",
        "rationale": "The discontinuation rate associated with this oral dosing regimen was
        "studyAcronym": "LZZT",
        "type": {
          "id": "Code_1",
          "code": "C98388",
          "codeSystem": "http://www.cdisc.org",
          "codeSystemVersion": "2023-09-29",
          "decode": "Interventional Study"
        },
        "studyPhase": {
          "id": "AliasCode_1",
          "standardCode": {
            "id": "Code_2",
            "code": "C15601",
            "codeSystem": "http://www.cdisc.org",
            "codeSystemVersion": "2023-09-29",
            "decode": "Phase II Trial"
          },
          "standardCodeAliases": []
        },
        "documentVersionId": "StudyProtocolDocumentVersion_1",
        "dateValues": [
          {
            "id": "GovernanceDate_1",
            "name": "D_APPROVE",
            "label": "Design Approval",
            "description": "Design approval date",
            "type": {
```

Study

parent_entity	parent_id	parent_rel	rel_type	id	name	description	label	versions	documentedBy	documentedBy.id	
Wrapper		study	definition		Study_CDISC PILOT - LZZT			TRUE	TRUE	StudyProtocolDocument_1	Pro

StudyVersion

parent_entity	parent_id	parent_rel	rel_type	id	studyTitle	versionIdentifier	ra
Study		versions	definition	StudyVersion_1	Safety and Efficacy of the Xanomeline Transdermal Therapeutic System (TTS) in Patients with Mild to Moderate Alzheimer's Disease	2	The discontinuation rate dosing regimen was 58.6 alternative clinical strate improve tolerance for th development of a Transc (TTS) has been initiated.

Code

parent_entity	parent_id	parent_rel	rel_type	id	code
StudyVersion	StudyVersion_1	type	definition	Code_1	C98388
StudyVersion	StudyVersion_1	businessTherapeuticAreas	definition	Code_3	PHARMA

AliasCode

parent_entity	parent_id	parent_rel	rel_type	id	standardCode	standardCode.id	standardCode.code	standardCo
StudyVersion	StudyVersion_1	studyPhase	definition	AliasCode_1	TRUE	Code_2	C15601	http://www.
SubjectEnrollment	SubjectEnrollment_1	code	definition	AliasCode_2	TRUE	Code_67	150	ISO 3166-1 a

GovernanceDate

parent_entity	parent_id	parent_rel	rel_type	id	name	label	de
StudyVersion	StudyVersion_1	dateValues	definition	GovernanceDate_1	D_APPROVE	Design Approval	Design approva

CORE Report Generated by the Test Run

Conformance Details	
Report Generation	2024-03-08T14:15:29
Total Runtime	78.4 seconds
CORE Engine Version	0.6.2
Standards Details	
Standard	DDF
Version	V2
CT Version	
Define-XML Version	
UNII Version	Not configured
Med-RT Version	Not configured
MedDRA Version	Not configured
WHODRUG Version	Not configured
SNOMED Version	Not configured

CORE Report

- Generated in Excel
- Placed in the CORE folder
- Datetime stamp in name

	A	B	C	D
1	Dataset	CORE-ID	Message	Issues
2	BiomedicalConcept	CORE-000424	The combination of Code and CodeSystem for a set of instances of a relationship from a Class to the Code class includes duplicates	41
3	BiomedicalConceptProperty	CORE-000408	A required BiomedicalConceptProperty (isRequired=true) is not enabled (isEnabled=true), or a disabled BiomedicalConeptProperty (isEnabled=false) is required (isRequired=true)	1
4	BiomedicalConceptProperty	CORE-000424	The combination of Code and CodeSystem for a set of instances of a relationship from a Class to the Code class includes duplicates	120
5	Code	CORE-000427	Within a submitted StudyVersion, multiple codeSystemVersions are used for the same codeSystem, and/or multiple decodes are used for a the same code.	46
6	Encounter	CORE-000414	The relationship links an entity instance to itself (previousId, nextId or defaultConditionId is the same as id), which is invalid	10

Rule examples

- Schema
  - Model: data types and relationships according to json API specification
  - References:
    - Previous/next ordering of encounters, eligibility criteria, epochs etc
    - All specified BCs, procedures, categories referenced by activity
    - Target references exist.
- Conditional
  - Attributes/Content:
    - fully defined timing window
    - At least 1 address field specified
  - Content:
    - Subject enrolment units
    - isRequired and isEnabled dependency
    - 1 objective with level primary
    - Age, gender, enrolment number etc specified at either cohort or study population level

	A	B	C	D	E
1	Check_ID	State	Check Category	Short check Description	Affected Classe(s)
25	CHK0024	UPDATE	CONDITIONAL	Either defaultCondition or TimelineExit should be filled	ScheduledActivityInstance, ScheduledDecisionInstance
26	CHK0025	FINAL	CONDITIONAL	isRequired and isEnabled dependency	BiomedicalConceptProperty
27	CHK0026	FINAL	CONDITIONAL	CDISC Study Phase codes used	StudyVersion
28	CHK0027	> V2.6	CONDITIONAL	At least 1 applyToIds specified for condition	Condition
29	CHK0028	FINAL	CONDITIONAL	If studyAmendement Other then reasonOther specified and vice versa	StudyAmendmentReason
30	CHK0029	FINAL	CONDITIONAL	Timing Anchor should be Start to Start	Timing
31	CHK0030	FINAL	CONDITIONAL	At least 1 main timeline for a studyDesign	ScheduleTimeline
32	CHK0031	> V2.6	CONDITIONAL	At least 1 anchor per timeline	ScheduleTimeline, Timing

Rule examples

- Conditional
  - Complex:
    - Unique arm-epoch combination
    - Study type matches number of interventions
    - Alignment between encounter and epoch ordering with timeline
- Unique
  - Ids: within and between classes
  - Codes:
    - Only 1 entry per title type, signature type etc
    - No repeats of codes for 1 entry
  - Text:
    - Name description label within a class
    - Code system version
    - Section numbers

A	B	C	D	E
Check_ID	State	Check Category	Short check Description	Affected Classe(s)
CHK0099	CREATE	CONDITIONAL	If study type=INTERVENTION then study intervention referenced from a procedure	Procedure, StudyIntervention
CHK0100	FINAL	SCHEMA	A scheduledInstance should not point to it's own timeline	scheduledActivityInstance, scheduledDecisionInstance, scheduleTimeline
CHK0101	FINAL	UNIQUE	Within a StudyVersion, a codeSystem, and a codeSystemVersion, code and decode must be one-to-one.	code
CHK0102	DEFINE	CONDITIONAL	A sponsor identifier is specified	StudyIdentifier
CHK0103	DEFINE	SCHEMA	Referred TimelineExits in the scheduledActivityInstance class defined in the corresponding ScheduleTimeline class	scheduledActivityInstance, scheduledDecisionInstance, scheduleTimeline
			Referred entryIds in the	

CORE rule features used for USDM

- Existing rule features like:
  - Equal_to
  - Does_not_contain_case_insensitive
  - Is_not_unique_set
- New rule features created like:
  - Enable test data format
  - Joining datasets
  - …
- New rule features still to be created
  - JSON schema check
  - XML format check
  - Links to DDF/external codelists
  - Complex joins and cross checks
  - … ?

cdisc

- **Planning for Phase 4**

Scope of Phase 4

- USDM version 3.0 final set of rules
- Add new YAML functionality to enable all anticipated rules like for:
  - Complex cross-checks
  - Check CT
  - Null values checks
  - API Json schema check
  - XML format check
- Create improved reporting template
- USDM version 4.0 final set of rules
- Add an exemplar study

cdisc

Q&A