

CDISC Library try-out: from implementation to evaluation of the API

Roman Radelicki, SGS, Mechelen, Belgium

ABSTRACT

CDISC Library promises linked data and a REST API to deliver CDISC standards metadata for software applications to automate standards-based processes. With my background as a programmer, my curiosity was immediately triggered during the presentation on the CDISC 360 project: a REST API (Application Programming Interface), linked metadata and biomedical concepts!

In this paper, I want to share the journey of the CDISC Library implementation in one of our in-house developed tools. To test and demonstrate the implementation of the CDISC Library, we used our TS (Trial Summary Information) dataset creation tool. This tool was the ideal case study candidate because it is a standalone tool which makes use of the CDISC SDTMIG and CDISC CT data, without a major impact on other tools allowing quick deployment of the CDISC Library. The TS dataset creation tool enables the Clinical Data Manager to create the TS dataset for a trial, necessary for submission to the authorities. The Clinical Data Manager is able to add trial summary parameters which are based on the CDISC controlled terminology.

In this paper, I will explain the tool and handle the API calls in more detail. My thoughts and findings on the use of CDISC Library within the tool will be shared. I will compare the CDISC Library to the former way we handled the CDISC SHARE data. Finally, I will also evaluate what might still be missing and future opportunities for improvement.

INTRODUCTION

What is the CDISC 360 project? CDISC 360 aims to support standards-based, metadata-driven automation across the end-to-end clinical research data lifecycle. The standards we download and use from the CDISC Library Archives (formerly known as CDISC SHARE) contain more text than metadata. Gaps in standards metadata limit automation opportunities. Today CDISC provides normative structural description of data (2 dimensional tables – columns) we use to create our datasets; what we are missing are semantics and relationships between information. Their goal is also that non-standard experts can use the CDISC standards. CDISC will add computer readable process metadata which enables end-to-end automation and will evolve from normative to informative standards. CDISC 360 will develop concept-based standard definitions, and test and demonstrate end-to-end automation of study specification and data processing during 18 months by conducting three use cases.

Metadata Centric Approach

A metadata driven approach and the seamless integration of processes and people are key to creating clinical databases that are compliant and consistent with CDISC SDTM standards. The implementation of a metadata repository (MDR) was the first step toward a metadata driven approach; we at SGS went a few steps ahead of this and built in-house tools that are linked to the MDR to achieve end-to-end automation of the SDTM workflow. More information about our metadata centric approach can be found in this paper: “End to End SDTM Automation: A Metadata Centric Approach”

With every new release of CDISC standards, controlled terminology or client specific implementation of a standard, the data is fetched from the appropriate locations. Once downloaded or received, the files are imported into our MDR after which they become available and are ready to be used in all our linked tools and trials.

CDISC Library

It's clear from the CDISC 360 project that CDISC wants a different approach on how we will access and use the metadata. One of the changes which is already available to us is the CDISC Library.

From the CDISC website: CDISC Library uses linked data and a REST API to deliver CDISC standards metadata to software applications that automate standards-based processes. CDISC Library provides access to new relationships between standards as well as a substantially increased number of versioned CDISC standards and controlled terminology packages.

If we compare it to the SGS MDR approach above, it would mean that our software applications don't necessarily need to get their information from our metadata repository anymore, but can directly communicate with CDISC by making use of the API. (from Wikipedia: An application programming interface (API) is a set of subroutine definitions, communication protocols, and tools for building software) The application would always have access to the most recent up to date information available from CDISC without the need of first implementing a new standard.in our SGS maintained MDR.

Trial Summary tool

Please bear in mind that this case study only touches a fraction of the CDISC 360 concept and that its focus is on this level of SDTM and Controlled Terminology for the TS domain.

The ideal case study candidate to try out the CDISC Library API and the linked metadata is our Trial Summary datasets creation tool (referenced as TS tool from here on). Unlike many other SDTM datasets, the data from the TS dataset is hard to retrieve programmatically from a single source. Therefore, it used to be created by the Clinical Data Manager in an excel file after which it was loaded into the database resulting in the TS dataset. Needless to say, this approach was prone to error. In first place, excel isn't the best format for exchanging data, with character mismatches and date format problems as a result. On the other hand, some values of Trial Summary parameters are linked with CDISC Controlled Terminology and therefore it is important correct values are used. To tackle these hurdles, we created our TS tool which is a module in our Data Entry Tool. We linked the tool to our MDR containing all metadata needed to make sure correct values are provided in the TS dataset. To demonstrate the use of the MDR in the TS tool, in the example below, the Clinical Data Manager will add a record in the TS dataset and select the correct Trial Summary Parameter (TSPARMCD) from a dropdown list (**Fig. 3**). The Values presented in the dropdown are coming from our linked MDR.

Variable Name	Variable Label	Type	Controlled Terms, Codelist or Format	Role	CDISC Notes	Core
TSPARMCD	Trial Summary Parameter Short Name	Char	TSPARMCD	Topic	TSPARMCD (the companion to TSPARM) is limited to 8 characters and does not have special character restrictions. These values should be short for ease of use in programming, but it is not expected that TSPARMCD will need to serve as variable names. Examples: AGEMIN, AGEMAX	Req

Fig. 1 TSPARMCD from the SDTM Implementation Guide v3.2

Code	Codelist Code	Codelist Extensible (Yes/No)	CDISC Submission Value	CDISC Synonym(s)
C66738		Yes	TSPARMCD	Trial Summary Parameter Test Code
C98703	C66738		ACTSUB	Actual Number of Subjects
C146995	C66738		ADAPT	Adaptive Design
C49703	C66738		ADDON	Added on to Existing Treatments

Fig. 2 excerpt from the TSPARMCD codelist in the Controlled Terminology 2019-12-20

Fig. 3 Integration in our TS tool : dropdown list with available TSPARMCDs.

Fig. 1, 2 and 3 show a straightforward integration of the MDR in the TS tool. **Fig. 1** shows the SDTM Implementation Guide (SDTMIG) which indicates in the fourth column TSPARMCD is controlled by terminology: TSPARMCD. **Fig. 2** shows the controlled terminology in the excel file published by National Cancer Institute. And **Fig. 3** shows the integration in our TS tool when adding a record in the TS domain. Still some information seems to be missing. How do I know if a specific parameter is required or not for my trial?

Variable Name	Variable Label	Type	Controlled Terms, Codelist or Format	Role	CDISC Notes	Core
TSVAL	Parameter Value	Char	*	Result Qualifier	Value of TSPARM. Example: "ASTHMA" when TSPARM value is "Trial Indication". TSVAL can only be null when TSVALNF is populated. Text over 200 characters can be added to additional columns TSVAL1-TSVALn.	Exp

** Indicates variable may be subject to controlled terminology, (Parenthesis indicates CDISC/NCI codelist code value)*

Fig. 4 TSVAL variable from the SDTM Implementation Guide v3.2

There are some less straightforward situations. For instance, TSVAl. **Fig. 4** shows the TSVAl variable in the SDTMIG which indicates that the variable **may be** subject to controlled terminology. How would I know if a value is subject to controlled terminology? I cannot know from the SDTMIG metadata excel format downloaded from CDISC SHARE. Luckily more information is available in the PDF version of the implementation guide which is unfortunately not machine readable. In the file, there is an example with a table containing all the parameters that are required or expected in the Trial Summary dataset with an indication if TSVAl is subject to not only CDISC controlled terminology but also others like SNOMED, NULLFLAVOR (ISO 21090), UNII and more. If you are planning to submit to the FDA, you might want to check the Study Data Technical Conformance Guide where in appendix B (Trial Summary (TS) Parameters for Submission – Clinical) you can find another list of FDA desired Trial Summary parameters for submission. At the end of the SDTMIG PDF document there is also an appendix C1: Trial Summary Codes, with even more information about the parameters than in the example section. To address the issue of terms within codelists that may have relationships to other terms within other codelists, CDISC publishes Codetable mapping files which are more or less machine readable. After implementing this TS codetable mapping file in our TS tool, we were able to link the values (TSVAl) to the corresponding parameters (TSPARMCD). **Fig. 5** illustrates the dropdown list in the TS tool the user is presented with when selecting the TSVAl column for the TSPARMCD = ADAPT.

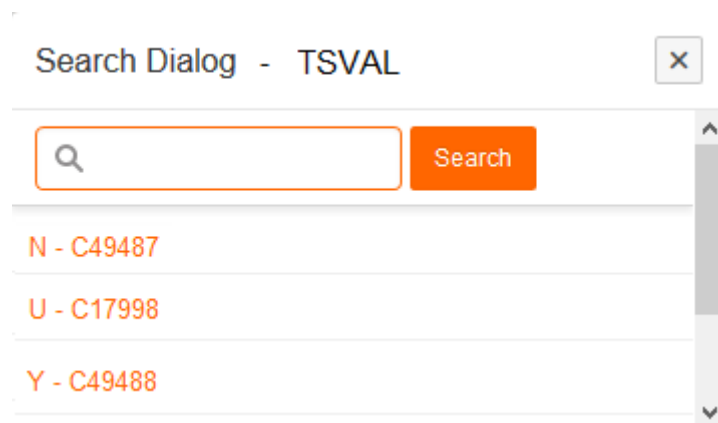


Fig. 5 TSVAl dropdown list for TSPARMCD = ADAPT

All the information to link the variables with the controlled terminology is out there, but you will have to find your way through several locations and documents. These examples show the gap in the metadata and makes automation less straightforward. This is exactly what CDISC wants to fix.

With the release of the CDISC Library, did they already address these issues? Do they really provide linked metadata?

CDISC LIBRARY API

To find out more about CDISC Library, I started with the CDISC website. Everything we need for our TS tool should be available in the CDISC Library: SDTM, SDTMIG and Controlled Terminology.

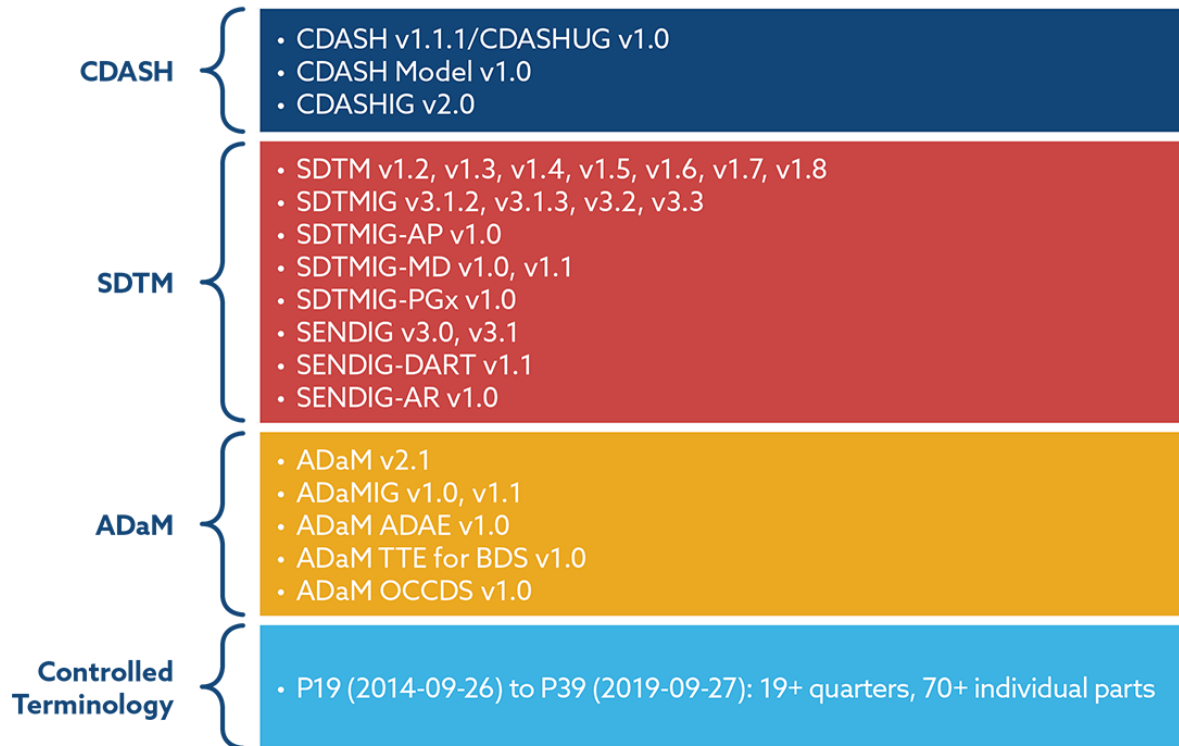


Fig. 6 CDISC Library available CDISC standards and terminology. Taken from the CDISC website.

The next step was to get an account, which for a CDISC member went very smoothly. This account is needed for authentication when performing API calls. Finally, I went through the online technical API documentation before creating my first CDISC Library API call.

As starting point I will take the example from above and try to fetch all the codelist information for the TSPARMCD variable to be able to populate the dropdown list in our TS tool as shown in **Fig. 3**

First, I will create an API request to get more information about the TSPARMCD from the SDTM Implementation Guide. I will use xml to share the response I get from the API call.

```
GET /mdr/sdtmig/{version}/datasets/{dataset}/variables/{var} Get SDTMIG Dataset Variable
```

Fig. 7 documentation from CDISC about the API call to retrieve information about an SDTMIG variable

{version} is the SDTM Implementation Guide version used. For this example, I will use SDTMIG v.3.2

{version} = 3-2

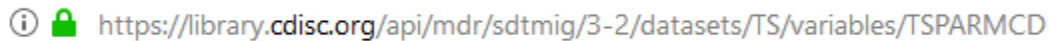
{dataset} is the domain in which the variable of interest exists. In this example TS

{dataset} = TS

{var} is the variable of interest. In this example TSPARMCD

{var} = TSPARMCD

This will result in following API request which is called within our TS tool:



<https://library.cdisc.org/api/mdr/sdtmig/3-2/datasets/TS/variables/TSPARMCD>

Fig. 8 API request for variable TSPARMCD

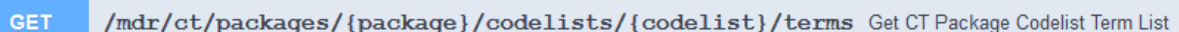
This is the XML response from the above (Fig. 8) API request:

```
- <datasetVariable>
  <ordinal>5</ordinal>
  <name>TSPARMCD</name>
  <label>Trial Summary Parameter Short Name</label>
  + <description></description>
  <role>Topic</role>
  <simpleDatatype>Char</simpleDatatype>
  <core>Req</core>
- <_links>
  + <self></self>
  - <codelist>
    <href>/mdr/root/ct/sdtmct/codestats/C66738</href>
    - <title>
      Version-agnostic anchor resource for codelist C66738
    </title>
    <type>Root Value Domain</type>
  </codelist>
```

Fig. 9 XML response from the TSPARMCD API call

The XML response provides us with the SDTMIG TSPARMCD information. The part in green in Fig. 9 gives more information about the SDTMIG variable requested. (Some additional information was removed from this example to save space in the screenshot.) The orange part is what is of interest, the codelist attached to the TSPARMCD variable. The yellow part is the linked metadata CDISC is talking about. This link will provide us with the information we need about the codelist.

This brings us to the next API request we need to perform to retrieve the links to the terms in the TSPARMCD codelist



GET /mdr/ct/packages/{package}/codestats/{codestats}/terms Get CT Package Codestats Term List

Fig. 10 documentation about the API call to retrieve all the terms from a codelist

{package} is the controlled terminology version used. For this example, I will use SDTM Controlled Terminology from June 28th, 2019

{package} = sdtmct-2019-06-28

{codestats} is the C-code of the codestats of interest. In this example the code retrieved in Fig. 9 for TSPARMCD (C66738)

{codelist} = C66738

This will result in the following API request:



Fig. 11 API request for codelist C66738

This is the XML response from the API request (**Fig. 11**):

```
- <cdiscLibrary>
  - <codelist>
    <conceptId>C66738</conceptId>
    <extensible>true</extensible>
    <name>Trial Summary Parameter Test Code</name>
    <submissionValue>TSPARMCD</submissionValue>
    - <term>
      - <href>
        /mdr/ct/packages/sdtmct-2019-06-28/codelists/C66738/terms/C98703
      </href>
      <title>Actual Subject Number</title>
      <type>Code List Value</type>
    </term>
    - <term>
      - <href>
        /mdr/ct/packages/sdtmct-2019-06-28/codelists/C66738/terms/C146995
      </href>
      <title>Adaptive Study Design Indicator</title>
      <type>Code List Value</type>
    </term>
```

Fig. 12 XML response from the API request

The part in green in **Fig. 12** gives more information about the codelist requested. (Some additional information was removed from this example to save space in the screenshot.) What follows after the green part are all the terms in the TSPARMCD codelist but not yet the values of the TSPARMCDs we are looking for. In this example only two terms are shown: Actual Subject Number and Adaptive Study Design Indicator marked in orange. The yellow part is again the linked metadata CDISC is talking about. This link will provide us with the information we need about the codelist.

The next API requests we need to perform are all the links to the terms provided in the call above. As an example, I will take the Adaptive Study Design Indicator C146995

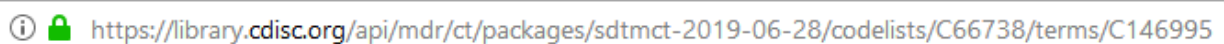


Fig. 13 API request URL for codelist C146995

This is the XML response from the API request (**Fig. 13**):

```

- <discLibrary>
  - <term>
    <conceptId>C146995</conceptId>
    <submissionValue>ADAPT</submissionValue>
    - <definition>
      Indicate if the study includes a prospectively planned opportunity for modification of one or more specified
      aspects of the study design and hypotheses based on analysis of data (usually interim data) from subjects in the
      study.
    </definition>
    <preferredTerm>Adaptive Study Design Indicator</preferredTerm>
    <synonyms>Adaptive Design</synonyms>
    + <_links></_links>
  </term>
</discLibrary>

```

Fig. 14 XML response of the API request for codelist C146995

The XML response (**Fig. 14**) provides us with the TSPARMCDs which we can use in the TS tool. To find out the correct label for the parameter, we should perform the same actions, but we should start from the TSPARM codelist instead of the TSPARMCD.

CDISC really delivers linked metadata. I was able to find my way from the SDTMIG TS dataset to the codelist used for TSPARMCD by making use of the API calls. Having the controlled terminology linked with the SDTMIG metadata is already a big benefit. But what about those less straightforward situations like the TSVAl in **Fig. 4** which may be subject to controlled terminology; was this addressed in the API? At the moment of writing and to my knowledge, the answer is unfortunately 'No'. I have examined the XML response for the TSVAl variable, with no trace of any reference to controlled terminology. According to the Codetable mapping file for TS, the value one should provide for the ADAPT codelist (C146995) in the example above (**Fig. 14**) is actually controlled by terminology C66742 No Yes response, but this is not shown in the XML response.

CONCLUSION

CDISC has really chosen the path to the future with their linked metadata and concept-based standard definitions. This case study of implementing the CDISC Library API calls in a small stand-alone software tool like the TS tool, has proven that we are not quite there yet. The information retrieved from the old CDISC SHARE versus the information pulled from the API is not that different from the 2-dimensional data that we know today. At this moment, the CDISC Library provides additional traceability concerning versioning, but more linked metadata doesn't seem to be provided. This means that, by only making use of the CDISC Library API calls, we are not yet able to provide the same functionalities in our TS tool. We still need to implement the codetable mapping file to be able to retrieve the correct values that should be used in TSVAl. Nevertheless, I strongly believe CDISC has laid the foundations for a bright future for automated standards-based processes. The more linked metadata is provided in the CDISC Library, the better software it will deliver.

Another consideration to make is the way the API is implemented in software. In the TS tool case study, we decided to directly call the API within the tool. This means that for the TS tool we are dependent on the availability of the CDISC API. Until now, we were completely dependent on the availabilities of in-house developed systems. With the use of the CDISC Library API however, next to our own systems, we need to take into account the CDISC's availability e.g. downtimes due to maintenance or upgrades. On the other hand, calling APIs directly from CDISC will provide us with the most up to date information available without having to constantly implement every new release at our side. Thus, creating

automation and reducing our workload. This will also come in handy later on when CDISC keeps its promise to increase the number of versioned CDISC standards and controlled terminology packages.

REFERENCES

CDISC SDTM Controlled Terminology: <https://www.cancer.gov/research/resources/terminology/cdisc#1>

Codetable Mapping Files: <https://www.cdisc.org/standards/terminology>

SDTM Implementation Guide: <https://www.cdisc.org/standards/foundational/sdtmig>

CDISC 360 slide deck:

https://www.cdisc.org/system/files/all/CDISC_360/CDISC_360_Project_Description.pptx

SGS paper: End to End SDTM Automation: A Metadata Centric Approach:

<https://www.lexjansen.com/phuse-us/2019/si/SI01.pdf>

Wikipedia: https://en.wikipedia.org/wiki/Application_programming_interface

CDISC Library API: <https://www.cdisc.org/cdisc-library>

CDISC Library API technical documentation: <https://www.cdisc.org/cdisc-library/api-documentation>

FDA Study Data Technical Conformance Guide v4.4: <https://www.fda.gov/media/131872/download>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

clinicalresearch@sgs.com

EUROPE: +32 15 27 32 45

AMERICAS: + 877 677 2667

WWW.SGS.COM/CRO

Brand and product names are trademarks of their respective companies.