PharmaSUG 2018 - Paper SS-14

Diagnostics of technical errors in define.xml file

Sergiy Sirichenko, Max Kanevsky, Pinnacle 21

ABSTRACT

Study metadata in standardized format plays critical role in automated processes. Wrong or missing information and technically invalid representation of define.xml file may result in incorrect processing of submission data at FDA and PMDA.

Today there is a lack of understanding in what study metadata is critical for automated processes and how to ensure its correct implementation. Invalid technical implementation of Define-XML standard and additional regulatory requirements may lead to major errors in metadata structural consistency and missing information. Pinnacle 21 software helps identify these errors, but some validation messages may be tricky and hard to interpret by less experienced users.

This paper will identify the most important study metadata utilized by automated processes and show examples of errors caused by missing or incorrectly implemented define.xml file. This paper will also provide guidance on how to perform diagnostics of technical errors in define.xml file..

INTRODUCTION

Why define.xml is important?

CDISC standards are required for submitting study data to FDA and PMDA, as they enable the use of standard-based review and analysis tools, automating and speeding up the review process. Until recently regulatory review was a manual process with limited need for data standardization, which is why its enforcement has been minimal. The automation of regulatory review is a game changing event with an immense impact on how industry should approach data standardization and submission preparation.

Automation means that processes like uploading data into clinical data repository or executing standard analysis are done without any human involvement. In addition to standardized data and use of standard tools, a process configuration step is also automated.

Define.xml file plays a critical role in automating processes as a major source of machine-readable study metadata. Some small errors in define.xml file may be critical for executing automated processes.

For example, when you run validation of SDTM data using Pinnacle 21 tool, you specify standards info manually. FDA DataFit (customized Pinnacle 21 Enterprise) extracts a version of SDTM standard from define.xml file and uses it for validation.

```
def:StandardName="SDTM-IG"
def:StandardVersion="3.1.2">
```

If this version is incorrect, then a validation process will produce both false-positive and false-negative results.

PMDA requires sponsors to fix validation Reject issues and explain all validation Errors. There are special PMDA consultation meetings with sponsor during submission process to reconcile sponsor's issue explanations and validation results independently reported by PMDA's installation of Pinnacle 21 Enterprise. Unexpected false-positive and false-negative validation results due to incorrect study metadata in define.xml file may delay submission process. [1]

If study metadata is invalid or missing then some compensatory actions are taken. For example, the latest version of a standard or dictionary will be utilized instead. It may result in the same unexpected outcomes of automated processes as described above.

For example, FDA DataFit uses define.xml file as a machine-readable source of MedDRA version.

```
<ExternalCodeList Dictionary="MedDRA" Version="19.0"/>
```

If this information is missing, then the latest version of MedDRA is utilized instead.

Incorrectly implemented (e.g., like Comment instead of Dictionary) MedDRA version also means missing machine-readable metadata.

Missing decimal points in MedDRA version (e.g., “19” instead of “19.0”) is PMDA Rejection criteria. Such issue must be fixed by Sponsor before proceeding further. [2]

There are still some limitations in data standards to fully support automated data processes at the regulatory agencies. For example, today there is no machine-readable info about CDISC Control Terminology (CT). Therefore, the latest version of CT is utilized by FDA DataFit for validation regardless of version used for studies. The industry is waiting for release and adoption of new Define-XML v2.1 standard, which can specify version of CDISC CT.

LOGIC OF DEFINE.XML VALIDATION

Diagnostics of validation messages for define.xml file may be tricky and requires good knowledge of validation logic and computational algorithms for tool-specific implementation of business rules.

Define-XML standard was created as an extension of CDISC Operational Data Model (ODM) standard. In addition to Define-XML specifications, CDISC published XML schema with expected structure for define.xml file. This XML schema is machine-readable specification for Define-XML, which can be utilized by generic XML tools for partial validation of define.xml files.

However, there are other validation rules for define.xml file in addition to XML schema. For example, Pinnacle 21 has 27 XML schema related checks and 96 additional content-related checks like submission-specific business rules or confirming correct implementation of CDISC Control Terminology (CT). For example, an additional requirement that a value for MedDRA version should include decimal point is a PMDA Rejection criteria for submission study data. Therefore, it is important to ensure that all validation rules for define.xml file were executed during validation.

There are three general steps of define.xml validation. First, it’s necessary to ensure that there are no major structural problems in define.xml file and its content is readable and can be upload for the next step of validation. If the first step failed, then content related checks cannot be executed or may produce unexpected results. Tuning validation algorithms to handle common errors in structural consistency of define.xml files are a continuous process. However, it is still not perfect to deal and diagnose new unexpected cases.

After fixing all technical and content issues in define.xml file, an additional and usually separate third step of validation is required to ensure consistency of study data and metadata stored in define.xml file. For example, when using Pinnacle 21 Community a user should execute two separate validations. One validation for define.xml itself (Figure 1) and other validation for study data with inclusion of define.xml file (Figure 2).

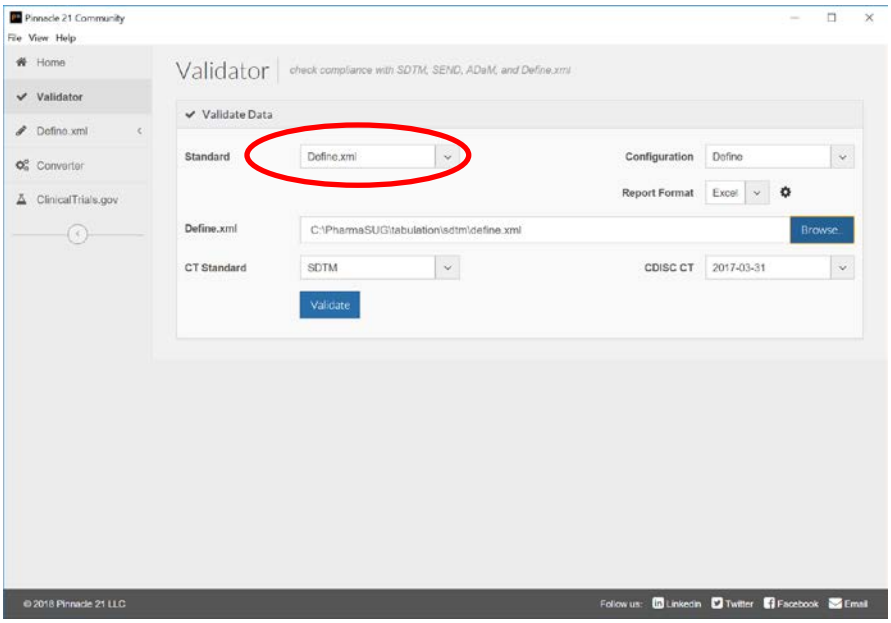

Figure 1. Validation of structure and content of define.xml file

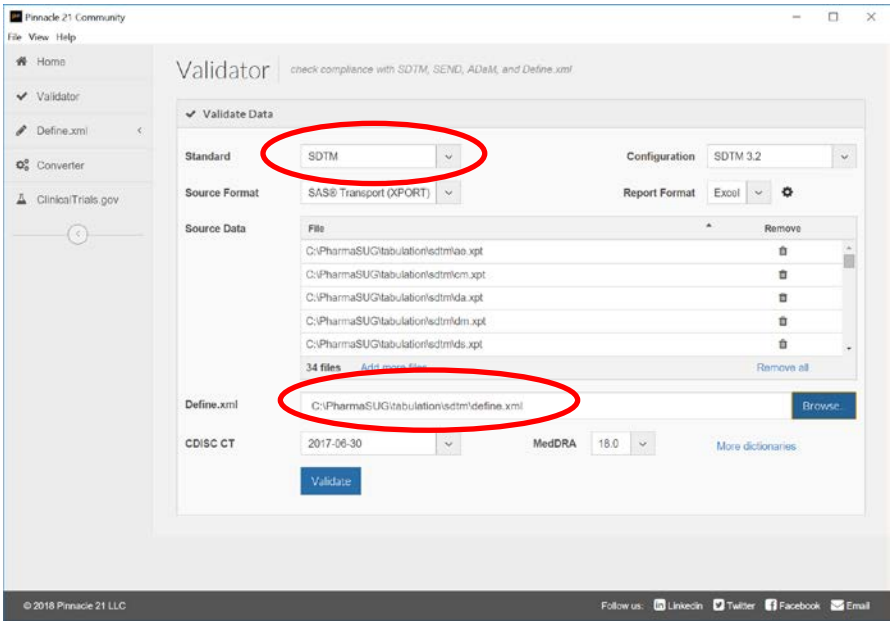

Figure 2. Validation of study data including its consistency with define.xml file

CONFIGURATIONS FOR DEFINE.XML VALIDATION

Configurations for define.xml validation is taken from users input in GUI/CLI and from define.xml file itself (Figure 3). FDA automated user input part and completely relies on metadata in define.xml file.

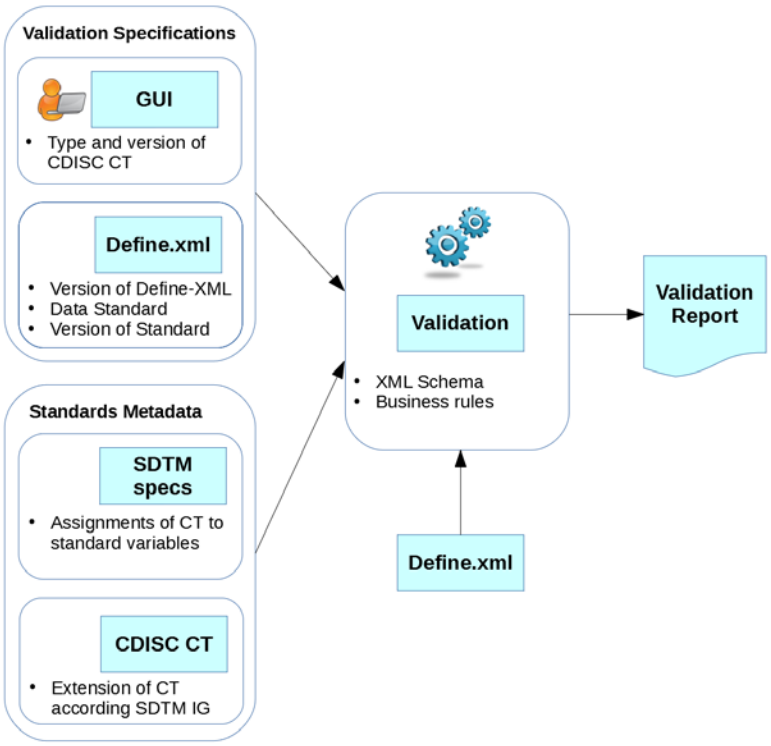

Figure 3. Configurations for define.xml validation

Manual input specifies type of CDISC CT (e.g., SDTM or ADaM) and its version. Also, the validator uses a location of define.xml file as a home folder for all files referenced in define.xml like annotated CRFs, datasets and other.

Then the validator starts reading of define.xml to identify a version of Define-XML standard as well data standard and its version which will be used for validation.

```
<MetaDataVersion OID="MDV.CDISC01.SDTMIG.3.1.2.SDTM.1.2"
  Name="Study CDISC01, Data Definitions"
  Description="Study CDISC01, Data Definitions"
  def:DefineVersion="2.0.0"
  def:StandardName="SDTM-IG"
  def:StandardVersion="3.1.2">
```

Data standard name (e.g., “`SDTM-IG`”) and its version (e.g., “`3.1.2`”) is used to ensure compliance of study metadata with standards.

Also, SDTM, SEND and ADaM standards’ specifications include assignments of CDISC CT codelists to standard variables. Validation utilized version of CDISC CT specified in GUI/CLI input.

Pinnacle 21 Control Terminology files are based on ODM files available on NCI website [3]. However, they include extensions according to additional specifications in SDTM/SEND/ADaM Implementation Guide documents. For example, Baseline Flag variables may be populated only with a single value ‘Y’ from standard ‘Yes/No’ Codelist in CDISC CT which includes 5 terms instead of 1 and therefore cannot be directly utilized for validation of CT for Baseline Flag variables.

If there is something wrong with validation configuration specifications, then validation process may produce unexpected or confusing results which require special diagnostics based on good knowledge of validation logic and computational algorithms for validation checks.

For example, if version of utilized Define-XML standard is missing or provided as invalid value (`def:DefineVersion="1"` instead of correct `def:DefineVersion="1.0.0"`), version 2.0 will be utilized for validation as the most recent version of Define-XML standard.

COMMON CONFUSING VALIDATION MESSAGES

Here are some examples of common and potentially confusing validation messages. Most of them are due to issues which are related to validation configuration stage and lead validation process into the wrong direction.

VALIDATION ERRORS DUE TO INVORRECT VERSION OF CDISC CT

Sometimes this issue is reported for valid standard terms and their NCI codes. In such cases one of the possible explanations is an incorrect version of CDISC Control Terminology used for validation.

Here is an example. Validation of define.xml file in pre-clinical study identified an Error DD0028: *Term/NCI Code mismatch in Codelist 'Laboratory Test Code'*. A reported term '*CALB'* and its NCI Code '*C125942'* are actually correct standard term and its corresponding NCI Code in CDISC SEND Control Terminology.

The actual problem is that this term was introduced in CDISC SEND CT version 2016-03-25, while according to sponsors Reviewer's Guide, CT version 2015-12-18 was utilized in the study. A validation was performed based on study metadata provided by sponsor.

Pinnacle 21 was configured to use SEND CT 2015-12-18 and utilized this standard metadata as a lookup table to find a match for '*CALB / C125942'* which does not exist yet in this version of SEND CT.

Therefore, this reported issue is actually due to inconsistency in versions of Control Terminology specified in study metadata and the one utilized for the study.

There are two options for resolving such issues:

1. Correct the version of CT in Reviewer's Guide and other related documents
2. Consider '*CALB'* term as non-standard according to originally claimed version of utilized CT; correct define.xml file by removing NCI Code for this term and flag it as a formal extension of standard CT

Note that standard terms may be different across versions of CDISC Control Terminology. For example, standard terms defined by NCI Code = *C17988* are used inconsistently across codelists and versions (Table 1).

Term	SDTM CT 2011-07	SDTM CT 2017-12
U	6	3
UNK	1	1
UNKNOWN	5	14

Table 1. Number of codelists representing unique terms for NCI Code = C17988 across different versions of CDISC SDTM Control Terminology

In particular, a term '*U*' was changed to a new standard term *'UNKNOWN'* in SDTM CT 2014-03-28 for 3 non-extensible codelists (ENRTPT), (STENRF) and (STRTPT).

This inconsistency in standard representation of the same information in CDISC CT may also result in DD0028 validation messages which may be confusing without understanding computational logic of define.xml validation in Pinnacle 21 tool.

Another case for confusing DD0028 validation message is due to use of standard terms from different CDISC CT codelists.

Development of CDISC tabulation data standards (SDTM, SEND) does not have the same pace as fast evolution of CDISC Control Terminology. For example, in 2011 CDISC SDTM CT included 104 codelists, which were expanded more than times 7 to 730 codelists in 2017. There are many potentially useful codelists which are not formally assigned to any standard variables or value level items. There is missing guidance about intended application of many new codelists in CDISC CT. It may introduce potential confusion for programmers.

Here is another example from a pre-clinical study. Validation of define.xml file reported Error DD0028: *Term/NCI Code mismatch in Codelist 'PPSTUNIT'*.

The actual issue is the use of standard terms from a different CDISC CT codelist (PKUDMG), while referencing to CDISC CT codelist (PKUNIT).

'nmol/L/(mg/kg)' (C119458) is a standard term in CDISC CT (PKUDMG) 'PK Units of Measure - Dose mg', NCI Code = *C128685*. In define.xml this term is used in codelist with reference to CDISC CT codelist (PKUNIT) 'PK Units of Measure', NCI Code = C85494, which does not include this reported term.

CDISC SEND CT has 5 different Codelists for PK Units without explicit guidance about their usage (Table 2).

Codelist Code	Codelist Name	NCI Code	Number of Terms
PKUDMG	PK Units of Measure - Dose mg	C128685	144
PKUDUG	PK Units of Measure - Dose ug	C128686	127
PKUNIT	PK Units of Measure	C85494	292
PKUWG	PK Units of Measure - Weight g	C128684	56
PKUWKG	PK Units of Measure - Weight kg	C128683	54

Table 2. List of codelists representing PK Units in CDISC SEND Control Terminology 2017-12-22

Interesting that *'nmol/L/(mg/kg)'* (C119458) was a standard term in CDISC CT codelist (PKUNIT) 'PK Units of Measure', NCI Code = *C85494* in older versions until 2016-09-30. In this version of SEND CT, *'nmol/L/(mg/kg)'* term was removed from (PKUNIT) codelist and added into newly introduced codelists (PKUDMG) and (PKUDUG). It was added back to codelist (PKUNIT) in the most recent version 2017-12-22.

In this example, a term *'nmol/L/(mg/kg)'* is formally considered as a standard term associated with PPSTRESU variable and representing by CDISC CT (PKUNIT) codelist with an exception of time period from 2016-09-30 to 2017-12-22. Unfortunately, for this study sponsor utilized CT 2016-12-16 where a term *'nmol/L/(mg/kg)'* was treated as non-standard for CT (PKUNIT) codelist. A use of standard terms from different codelists does not allow formally treating these terms as standard ones within the original target CT codelist.

OTHER EXAMPLES OF ISSUES RELATED TO INVALID VERSION OF CDISC CT

Similar to previously described confusing validation message DD0028, there are other validation issues due to incorrectly provided version of CDISC CT utilized in the study. Here are some examples:

DD0032: *Missing NCI Code for Term in Codelist 'Unit'* with reported terms '*L/L', 'ng/mL', 'ng/mL/mg'*. These terms existed in CDISC SDTM CT 2014-03-28, but they were removed in more recent versions utilized for study.

DD0033*: Unknown NCI Code value for Codelist 'Specimen Material Type'*

DD0034: *Unknown NCI Code value for Term in Codelist 'Epoch'*

ISSUES DUE TO ADDITIONAL INVALID TERMS

The most common confusing validation message may be DD0024: *Invalid Term in Codelist 'No Yes Response'* reporting '*N*', '*U*' or '*NA*' terms which are standard terms in CDISC CT.

Actual issue is due to invalid utilization of non-relevant terms for *Flag* variables like DTHFL, --BLFL or --PRESP where these '*N*', '*U*' and '*NA*' terms are not applicable according to SDTM/SEND IG documentation.

SDTM *Flag* variables are assigned to CDISC SDTM CT codelist (NY) 'No Yes Response' which includes 4 terms. However, SDTM IG specifies that these variables may only have either '*Y*' or a missing value. For validation purpose, original NCI files of CDISC CT are extended by Pinnacle 21 with additional codelists including a subset of (NY) codelist limited to a single term '*Y*'.

Variable codelist in a define.xml file may have a reference to CDISC CT specific codelist. To avoid potential mistakes done by programmers in assignment of CDISC CT codelist to standard variables in define.xml file, Pinnacle 21 uses independent internal codelist assignment as a part of SDTM metadata. For example, if a codelist for LBTEST variable in define.xml has a reference to CDISC CT codelist (UNIT), the Validator will still use CDISC CT codelist (LBTEST) as specified by SDTM standard.

Some codelists in define.xml may be similar across many variables. Therefore, a single common codelist may be utilized for multiple variables instead of the same but independent codelists for each variable. (NY) *'No Yes Response'* codelist is an example of the most prevalent case.

A problem occurs when this codelist with multiple terms is incorrectly assigned to *Flag* variables. Validation of define.xml file reports this case as an invalid term in non-extensible CDISC CT codelist associated with Flag variables. However, there is no explicit reference to particular variables in addition to name of codelist and invalid terms in validation report. It makes diagnostics of DD0024 error more complicated.

Correct implementation of define.xml file should include separate codelists for each type of variables. Variable codelist should describe planned data collection process. Additional non-relevant terms in codelists may be confusing.

Another example is DD0024: *Invalid Term in Codelist 'Relation to Reference Period'* validation message due to a use of a shared codelist for --STRTPT and --ENRF variables. According to SDTM IG, --STRTPT variables may be populated only with terms '*BEFORE*', '*AFTER*', '*COINCIDENT*' and '*UNKNOWN*'('*U*' in recent versions of CDISC CT). They represent a subset of CDISC SDTM CT codelist (STENRF) 'Relation to Reference Period'. Therefore, all additional standard terms like '*DURING'*, '*DURING/AFTER'*, '*ONGOING'* are considered as invalid terms for --STRTPT variables. However, they may be used for --ENRF variables.

ISSUES DUE TO INVALID ASSIGNMENT OF CDISC CT CODELIST

Let's consider a case when Pinnacle 21 validation of define.xml file produces a message DD0028: *Term/NCI Code mismatch in Codelist 'UNIT'* reporting '*IU, C48579'.*

'*UI*' is a standard term in CDISC SDTM CT codelist (UNIT) 'Unit'. A provided NCI Code for this term is also correct. There is no issues around inconsistent version of CDISC CT.

An actual problem was due to invalid use of CDISC CT codelist (UNIT) for VSORRESU/VSSTRESU variables instead of CDISC CT codelist (VSRESU) 'Units for Vital Signs Results'.

As we have already mentioned above, the validator uses CDISC CT codelist assignments to standard variables as specified by CDISC standards. It ignores assignments of CDISC CT to standard variables in define.xml file as potentially unreliable information.

In this case, the tool uses CDISC CT codelist (VSRESU) as a lookup table for validation of '*IU, C48579'* combination. It exists in CDISC CT codelist (UNIT), but does not exist in expected correct codelist (VSRESU).

Note that this example, for the same reason, will also result in the similar issue DD0033: *Unknown NCI Code value for Codelist 'Unit'*.

ISSUES DUE TO INVALID DATA STANDARD AND INVALID VERSION

When validating define.xml file for ADaM data, you receive a message DD0045: *Missing Domain value*. However, a Domain attribute is applicable only for tabulation data and is not used for analysis data.

Knowing the validation logic, we can assume that there is something wrong with study data standards metadata in a define.xml file. This info from define.xml is used as configuration for validation. For example, if define.xml refers to SDTM standard, then all SDTM related business rules will be applied including a requirement for non-missing Domain values

```
def:StandardName="SDTM-IG"
def:StandardVersion="3.2">
```

Similar example is a validation message DD0055: *Invalid Class value.* However, Class values look good. What could be wrong with define.xml file?

The actual problem is unsupported version of ADaM in define.xml file. Pinnacle 21 Community 2.2.0 does not support new versions of standards like ADaM IG 1.1 or SEND IG 3.1. If the tool cannot recognize expected versions of standards, then validation process may produce unexpected results.

```
def:StandardName="ADaM-IG"
def:StandardVersion="1.1">
```

Each validation report produced by Pinnacle 21 has a tab ‘Rules’. Refer to column Description of DD0021 and DD0022 rules for valid and supported Standard Names and their Versions.

FALSE-POSITIVE VALIDATION MESSAGES

Some validation messages may be false-positives like OD0012: *Invalid root element.* This rule has a description as ‘*Define.xml must contain a root element called ODM*’. However, XML code of define.xml file may look OK! with ODM root element present.

```
    <?xml version="1.0" encoding="ISO-8859-1"?><?xml-stylesheet
type="text/xsl" href="define2-0-0.xsl"?><!-- Produced from SAS data using
the SAS Clinical Standards Toolkit 1.6 --><ODM
xmlns:xlink="http://www.w3.org/1999/xlink" … >
```

This validation message is actually due to a bug in Pinnacle 21 Community 2.2.0. The Validator recognizes ODM element only if it starts with a new line of XML code like this modified code:

```
    <?xml version="1.0" encoding="ISO-8859-1"?><?xml-stylesheet
type="text/xsl" href="define2-0-0.xsl"?><!-- Produced from SAS data using
the SAS Clinical Standards Toolkit 1.6 -->
    <ODM xmlns:xlink="http://www.w3.org/1999/xlink" … >
```

It’s a common case observed in define.xml files created by SAS® Clinical Tool Kit and some other tools.

Often this issue is false-positively reported due to leading comments <!- …/> in front of ODM element.

```
    <!-- sample comments --><ODM … >
```

Until new version of Pinnacle 21 Community with this fixed bug will be available, you have a choice either to modify your define.xml file by starting ODM element with a new line or explain this validation message in Reviewers’ Guide as a known bug in Pinnacle 21 Community 2.0.0.

LINKS TO CRF PAGES DO NOT WORK

In addition to diagnostics of some confusing validation messages we would like to address common complaints from users that correctly implemented links to variable-specific CRFs pages in define.xml file do not work. Annotated CRFs document always opens on page 1 instead of a page specified by a link.

As the first step of diagnostics, you should ensure that stylesheet files for define.xml are not outdated. New version of define.xml stylesheet is available for download from CDISC website.

However, the most common source for this issue is related to settings of the browser used to view a define.xml file.

The simplest diagnostics is to try using different browsers and computers which may have different default settings. You need to ensure that your browsers (for example, Internet Explorer or Chrome) are configured to open linked *acrf.pdf* file within the browser rather than in other viewers like Adobe Acrobat Reader.

CONCLUSION

Define.xml file is a major source of machine-readable study metadata used by automated processes at FDA and PMDA. You should ensure that define.xml files include all expected content and do not have any technical errors. Understanding of define.xml validation logic is needed for diagnostics of validation findings.

REFERENCES

1. Mayumi Kominami, Takashi Kitahara, Yuichi Nakajima. “Hands on experiences of e-study data submission” CDISC Interchange Japan, 2017. Available at https://www.cdisc.org/system/files/all/event/restricted/2017_Japan/Japan%202017_Session%206_Kominami.pdf
2. Takuma Oda. “Experience of Electronic Study Data Submission” CDISC Interchange Japan, 2017. Available at https://www.cdisc.org/system/files/all/event/restricted/2017_Japan/Japan%202017_Session%206_Oda.pdf
3. CDISC Terminology. Available at https://www.cancer.gov/research/resources/terminology/cdisc

CONTACT INFORMATION

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

Sergiy Sirichenko
Pinnacle 21 LLC
1.570.817.6137
ssirichenko@pinnacle21.net

Max Kanevsky
Pinnacle 21 LLC
1.267.331.4431
mkanevsky@pinnacle21.net