PhUSE 2013

Paper CD09

Interpretation of the output from the OpenCDISC validator

Jørgen Mangor Iversen, LEO Pharma A/S, Ballerup, Denmark

ABSTRACT

Although the OpenCIDISC validator is a wonderful tool to reinforce compliance to standards, the reports generated can be complex and ambiguous, and thus open to interpretation. To begin with, the standards are ambiguous and contradictory; the OpenCDISC organization adds its own contradictions; and your company may have its own interpretation of the standards. Subsequently we must be able to determine our selves which elements of the report we need to act upon, and which we can safely ignore. Knowing where the ambiguities are, helps us distinguishing between important findings that disclose real errors in our data and the mere noise generated by the tool.

INTRODUCTION

The OpenCDISC validator is a free, open source, fully configurable tool for validating SDTM, SEND and ADaM data standards [0] provided by the consultancy company Pinnacle 21 in Pennsylvania, USA, of unknown ownership and dependencies. Trust in the product must come from the open source nature, even though the forum members don't seem to have access to contribute to the code, possibly explaining the low activity of the forum, but more so trust comes from the experienced quality of the product.

The need for something like the OpenCDISC validator stems from having seen many imaginative interpretations of the CDISC [2] standards, both from in-house Data Management departments and from external CROs. The notion of having international standards for clinical (or any other) data ought to ensure compatibility and interoperability of data, but the tendency seems to be to bend the standards to encompass everyone's company specific flavour. Several vendor products exist to validate CDSIC standards, and they are all probably perfectly fine, but the OpenCDISC validator is the one everybody can afford and is (in theory) vendor and data format independent [3].

The tool validates that clinical data in the forms of SDTM, SEND and ADaM comply with the CDSIC standards as interpreted by the OpenCDISC validator, within the limitations of the standards and the specific implementation of the standards. The tool validates both the formalities of the data (dataset names and labels, variable names, labels, type, format) as well as contents to some extent, particular cross references via foreign keys. These validation checks of the data ensures to a very great extent data consistency across studies and across sources of data.

CDISC AMBIGUITIES

The job of the OpenCDISC validator would be much easier, and the need for it equally less, if the CDISC standards were completely unambiguous. This paper will focus on the SDTM 3.1.2 standard, as this is the latest version accepted by the FDA (by 2013), and SDTM being the most important standard. However, all the CDISC standards are based upon ODM which has it own ambiguities such as the use of empty XML tags. Ambiguities have been identified in several places in SDTM 3.1.2, of which some might seem minor, but all becomes important when a computer has to deal with them.

- Variables --CAT, --DECOD, --LOC, --MODIFY, --SEV, TAETORD *label* vary between classes.
- Variables ARMCD, --DECOD, ETCD, IDVAR, IDVARVAL, TAETORD, --TOXGR, VISIT, VISITNUM *role* vary between classes.
- Variables AEREL, AESEV --BODSYS, --DECOD, COUNTRY, EPOCH, ETHNIC, IETEST RDOMAIN *Controlled Terminology* vary between different classes.
- Variable names reused for different purposes in different dataset as reflected by the variation of class and role.

The examples above are taken from the downloadable spread sheet [4] of the metadata from the CDSIC web site. These issues, and any other issues found, should be handled by defining unique variables for all uses and definitions. Only variables (such as keys and dates) that are identical in every aspect should have the same name. Otherwise they should be clearly different to enable machine readable data standards.

OPENCDISC AMBIGUITIES

It is quite obvious that the OpenCDISC validator (and everybody else) must make a few choices in order to interpret the ambiguities in the CDSIC standards. But in the process of doing so, the validator adds a little ambiguity of its own. That might be defended by the argument that the validation rules of the validator are fully configurable by any user. However, this only leads to foreseeable confusion about configuration files. It seems that it must be possible to have an unambiguous default configuration, when the supplied configuration file is as close to home as the case is. The process of validating clinical data runs so much smoother when there is not the question of configuration versions to worry about.

One ambiguity found is rules SD0026 [5] (Missing value for --ORRESU, when --ORRES is provided) and SD0029 (Missing value for --STRESU, when --STRESC is provided) which both require that units must always be present when original results and standard results respectively, are present. When collecting data as scores or ratios, or even calculating those and simply storing them in a findings domain, no units exist. If a missing unit was permissible for scores and, the validation report could be expected to yield zero messages and a perfect validation.

Another ambiguity is the difference between messages SD0017 'The value of Name of Measurement, Test or Examination (--TEST) should be no more than 40 characters in length' being a warning, and SD1049 'Qualifier Variable Label (QLABEL) value may have up to 40 characters' being an error. One could wonder why the wording is different and why one case is more severe than the other. Both messages result in truncation of data, and should thus be errors.

A STRATEGY FOR HANDLING MESSAGES

Wouldn't it be wonderful if a contract could be written that a CRO should deliver data that fully complied with standards and OpenCDISC validation yielded no messages at all? This is not very likely to happen even if standards and validation tools were perfect. The second best thing then becomes to define what level of messages is acceptable.

OpenCDISC validator messages comes in three severities; errors, warnings and informational. The distinction between these levels is not always logic. Some errors seem to be very harshly defined, such as certain values not in Controlled Terminology generate an error, while other values generate a warning. And some warnings seem a little lax, such as negative date intervals and references to non-existing visits. Informational messages are just that, and can always be safely ignored.

Experience shows, that raising the data to such a quality level that only errors regarding missing units on the likes of scores and ratios are generated by the validator, solves most data problems downstream when producing tables, figures and listing (TFL) and eventually submitting the data. However, this is not enough. Warnings of the category 'Cross-reference' should be treated with the same severity as errors, with possible exceptions being SD0065 'Invalid Subject Visit/Visit Number' and SD1023 'Invalid VISIT/VISITNUM'. Both these warnings are really indications of ambiguities of the standard, reflecting that one domain only captures planned visits, and another captures actual visits. Again, this is from the point of view of a computer trying to make sense of foreign keys and not having to deal with dangling references. And having said *that*, some of the 'Cross-reference' messages ought to be re-classified as 'Consistency', particular SD1014 'Invalid TAETORD' and SD1015 'Invalid EPOCH'.

And still this is not enough. Even some of the 'Consistency' warnings should be treated as errors. Examples here are:

- SD0040: Inconsistent value for --TEST within --TESTCD
- SD0046: Inconsistent value for QLABEL within QNAM
- SD0051: Inconsistent value for VISIT within VISITNUM
- SD0052: Inconsistent value for VISITNUM within VISIT
- SD0090: AESDTH is not 'Y', when AEOUT='FATAL'
- SD0091: AEOUT is not 'FATAL', when AESDTH='Y'
- SD1062: AESER is not 'Y', when AESOD equals 'Y'

All the messages above concerns 1:1 relationships between two variables, and all are prone to create duplicates when performing some data driven automatic processing of the data. In addition message SD1060 'Duplicate VISITNUM' concerns duplicates directly, and should be considered an error as well.

Taking all this into consideration a strategy for using the OpenCDISC validator as a tool for ensuring high data quality emerges, whether the source of data is external or internal:

- Aim for no errors, but allow for errors regarding missing units.
- Treat warnings classified as 'Cross-reference' as if they were errors, possibly allowing for invalid TAETORD and EPOCH messages.
- Treat the warnings classified as 'Consistency' and dealing with 1:1 relationships and duplicates as errors. Allow for no exceptions.

A strategy for handling messages should also contain a method for familiarise oneself with the many validation rules supplied with the OpenCDISC validator. One method is to view the XML configuration files supplied with the validator directly. They display very well in a browser, but gets organised per dataset, which can come in handy as a look-up document when constructing specific datasets. Another approach is to view any Excel report generated, switch to the 'Rules' sheet, apply filtering and segment by 'Category' and/or 'Severity'. This reduces the number of rules to small sets that may seem less overwhelming. In either case it makes sense to read each rule in its context and reflect on its implications towards the code for constructing datasets.

CONCLUSION

In spite of ambiguities of both standards and tools, the OpenCDISC validator is a great aid in ensuring high data quality. As it is still early days in the life cycle of standardising clinical data, it is a true pleasure to learn that the methods have such a mature status as the case is. It is to be expected that the handling of clinical data can be fully automated as standards and tools evolve. Understanding and analysing those data is a whole different story.

REFERENCES

1. OpenCDISC validator freely downloadable from http://www.opencdisc.org/.
2. Clinical Data Interchange Standards Consortium at http://www.cdisc.org/.
3. SAS version 5 transport file format (XPORT) definition can be found at http://support.sas.com/techsup/technote/ts140.pdf.
4. SDTM 3.1.2 metadata to be found in the member's area of http://www.cdisc.org.
5. All message codes taken from the 'Rules' sheet of OpenCDISC validation reports version 1.3.

CONTACT INFORMATION

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

Jørgen Mangor Iversen
LEO Pharma A/S
Industriparken 55
DK-2750 Ballerup
Denmark
+45 72 26 31 16
Joergen.iversen@leo-pharma.com
http://www.leo-pharma.com/