Paper TT08

Data Exchange in Clinical Trial – Programming Perspective

Chao Wang, FMD K&L, Fort Washington, USA
Bei Wang, FMD K&L, Fort Washington, USA
Yunsheng Wang, FMD K&L, Fort Washington, USA

ABSTRACT

Data exchange is heavily intertwined in the development and procedures of clinical trials. A large component of data exchange in clinical trials involves programming, and the perspectives of programming professionals. Programmers are responsible for maintaining Clinical Data Interchange Standards Consortium (CDISC), controlling terminology and cleaning related metadata for scientific and clinical use. Challenges that arise during the process of producing, managing and exporting the data include data accuracy and data security, among others. These challenges are made more difficult when working with multi-regional studies, different operating systems, and data obtained from different sources such as the Electronic Data Capture (EDC), Study Data Tabulation Model (SDTM) and Analysis Data Structure (ADaM). These data are finally compiled into clinical reports (tables, listings, and figures) that are then exchanged.

This paper will discuss the data encoding, translation, and transmission of "new" technologies, such as XPT, JSON, SSL, Blockchain and others, that have the potential to increase the flexibility and efficiency that programmers operate within the data exchange process. Finally, a new topic of stream data exchange and how it will reshape the future will be illustrated.

INTRODUCTION

Data exchange is defined as the manipulation of data from a source schema into a format that is compatible with a target schema [1]. It encompasses transferring study data between organizations and authorities, transmitting data between servers and computer programs, translating documents between different languages, and much more.

All of these processes involved in data exchange have to ensure that three criteria have been met: 1) the data is structured, 2) the data is an accurate representation of the source data, and 3) the data is comprehensively shared and compatible among different computer programs. By adhering to these criteria, researchers are able to use different data management technologies to improve data exchange efficiency and flexibility.

ANNOTATED CRF

When submitting a SDTM package to authorities, an annotated CRF (aCRF) is required. The aCRF is a file in PDF format with annotations that provide traceability between collected source data and SDTM domains/variables and ensure the SDTM is accurately representing the source data (Figure. 1).

Usually, programmers need to manually annotate each data point by following the metadata submission guidelines. This requires a tremendous amount of work; so, when an amendment just one-page long is provided, the complexity and the length of the project drastically increases. Annotations from one file may be copied to another; however, the annotation's location information is lost and must be manually redone. To make the work more efficient, Forms Data Format (FDF) can be used to import or export the interactive form information from a PDF. FDF is a file format for handling forms and annotations that are within PDF documents. In the FDF document using a text editor, one will find a catalogue object with the name FDF. The catalogue object contains a number of entries; the most important are Fields and F. Field points to the a list of input fields while F contains the filename of the PDF document, and based on this information you are able to modify fields with fdf_set_value(), fdf_set_opt() etc [2].

As shown in Figure. 2, by using FDF, programmers can share the annotations made across different file versions and are able to use coding to annotate the PDF directly in the FDF instead of manually annotate in PDF. This resolves the problem in a short period and make the data exchange process more efficient while ensuring the SDTM is an accurate representation of the collected data.

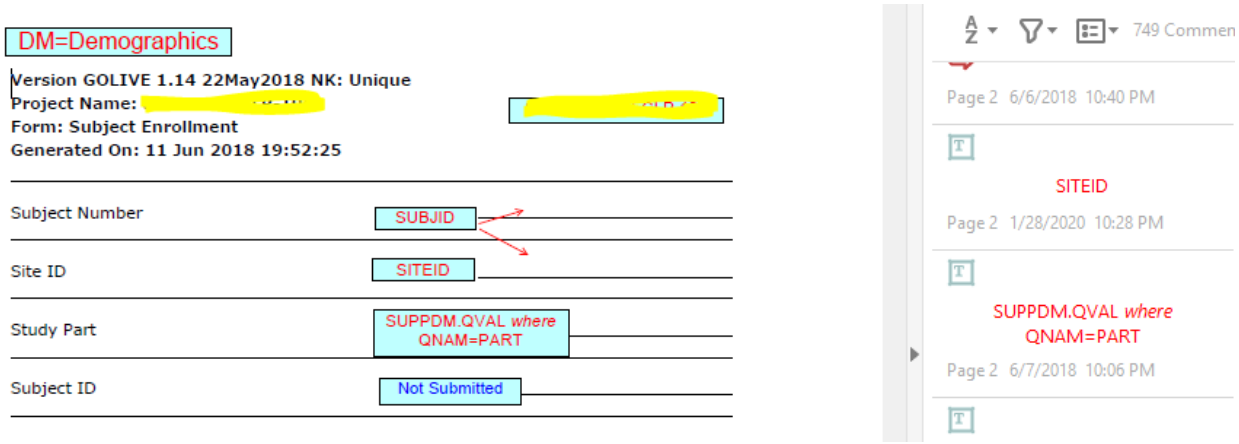


Figure 1 aCRF Sample (DM- SITEID)

```
2 0 obj
<</C[0.75 1.0 1.0]/Contents(SITEID)/CreationDate(D:20111213135654+08'00')/DA(0 0 0 rg /Helv 10 Tf)/DS(font:
Helvetica 10.0pt; text-align:center; color:#FF0000 )/F
4/M(D:20200128222826-05'00')/NM(e37caf04-e873-4c15-bc34-accd279c6ce3)/Page 1/RC(<?xml version="1.0"?><body
xmlns="http://www.w3.org/1999/xhtml" xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/"
xfa:APIVersion="Acrobat:10.0.0" xfa:spec="2.0.2"
style="font-size:10.0pt;text-align:center;color:#FF0000;font-weight:normal;font-style:nor\
mal;font-family:Helvetica;font-stretch:normal"><p dir="ltr">SITEID</p></body>)/Rect[306.489 575.96 367.978
590.163]/Subtype/FreeText/Type/Annot>>
endobj
```

Figure 2 FDF Sample (SITEID)

SAS TRANSPORT FILE

Structured data is powered by specific file formats. SAS Transport Format, version 5 (XPT) is a file format for submission of all clinical electronic datasets to FDA. As an open file format, XPT files can transfer between computer platforms or organizations. It can also convert to other commonly used formats such as CSV, SAS file [3]. However, this file format is highly used for almost 20 years without having been updated and making it hard to meet the new technical innovations and trends. Although SAS Institute published a new XPT version, SAS V8 Transport File Format, it can only be used on SAS 9.3. It fixed the length restriction on variables and related labels. However, there are still other limitations as shown below:

- Record layout is different between XPT v5 and v8, however, most of the programs cannot differentiate it through the file extension. If a program opened the wrong version, then it will not represent the data correctly.
- All character data are stored in ASCII, therefore, it only supports up to 256 western characters, and not support characters from other international languages.

There is no specific file extension to differentiate the XPT v8 and v5. A good practice is adding the number to the extension, like *.v8xpt. By using the macro LOC2XPT and XPT2LOC, SAS is one of the program that can differentiate the difference between XPT v5 and v8 by auto recognizing the characters in the file to determine which version should be applied.

To break the boundaries from SAS XPT, Dataset-XML provides a solution to transfer CDISC tabular data (SDTM, ADaM, SEND and etc.) in XML format across different entities. As per the implementation guide, the metadata for a data set contained within a CDISC Dataset-XML document must be specified using the CDISC Define-XML standard. JavaScript Object Notation (JSON) may be another solution. As an open-standard, language-independent file format, it is widely used with a diverse range of applications. CDISC API standard is implemented to JSON format now. These languages are independent and encode friendly with different standards in a variety of formats. They are able to facilitate the implementation of CDISC standards and ensure the data information is accurately shared between different organizations efficiently.

DATA EXCHANGE CHALLENGE

ENCODING

To guarantee the data can be shared accurately during the data exchange process, the encoding is the key.

Encoding translates the character sets to the codes that can be recognized by computer system. The well-known ASCII is a character encoding standard for electronic communication. It mainly supports 128 western characters and the extended ASCII set contains 256 characters [3]. The limited size of ASCII cannot support the requirement of using large set of characters in the global clinical trials. Per the latest FDA technical conformance guide, the provided data should avoid containing byte values from 160 to 191[4], as part of the character mapping in this range interferes with some agency processes. This means some of the commonly used symbols, such as °C or µ cannot be included in the datasets.

What if the EDC system truly collect the data that utilizes these symbols, what should we do? A universal encoding set is needed to solve this problem. Unicode is known as standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems and widely used in computing industry. Unicode can be implemented by different character encoding methods. UTF-8 is one of the most popular methods which is used in over 94% of websites on the internet as of November 2019 [5]. Most of EDC systems, like Medidata RAVA, provide data with UTF-8 encoding. SAS also provides the NLS (National Language Support) module to support the UTF-8 and various encoding sets.

If one uses the UTF-8 only for the data with extended characters, and mix the encoding during the delivery. The following ERROR would occur,

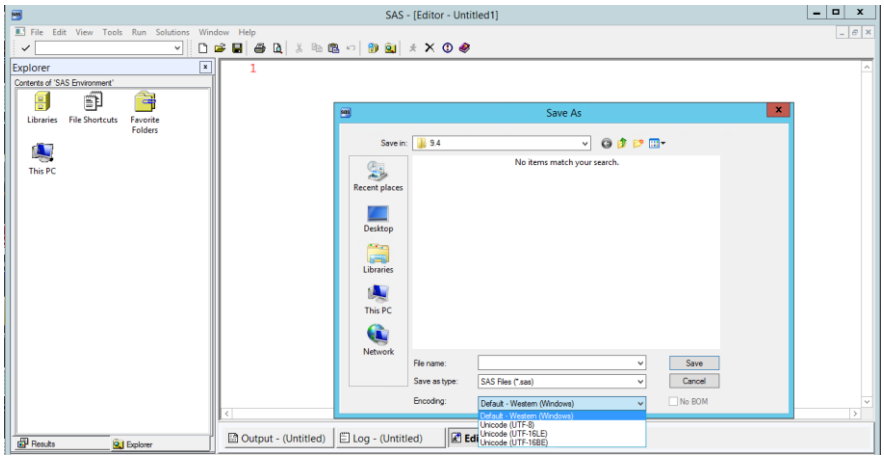
<sample1.png>

One of the main reasons is SAS manipulates the single byte character set (SBCS) and multiple byte character set (MBCS) in different ways. Like SAS functions, it has three levels: I18N Level 0, I18N Level 1 and I18N Level 2 (I18N is the abbreviation for internationalization). If there is any character requires multiple bytes to be saved, one should use the I18N Level 2 function in the whole package to ensure the correctly operation.

The encoding issue is not only related to the SAS datasets, it appears in all kinds of files. Like Define-XML, one should use UTF-8 encoding, instead of the default ISO-8859-1 encoding which was applied in almost all the Define-XML 1.0 version.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="define2-0-0.xsl"?>
```

Currently, the program codes are also required to be submitted along with the analysis datasets and reports. FDA guideline states the following. “Sponsors should submit software programs in ASCII text format” [5]. When one prepares the code, choosing the file encoding as UTF-8 and only contains the ASCII characters for US submission is preferred. The step to choose the encoding for SAS program files is as follow.



TRANSLATION

Encoding ensures the accurate representation. Furthermore, the consistent information needs to be shared with both human and machine during the data exchange. This is called Interoperability, which means the ability to

communicate and exchange data accurately, effectively, securely, and consistently with different information technology systems, software applications, and networks in various settings, and exchange data such that clinical or operational purpose and meaning of the data are preserved and unaltered [6]. There are three types interoperability:

- Technical Interoperability
- Semantic Interoperability
- Process Interoperability

A well-defined file format can help to achieve the technical interoperability. Standard metadata, dictionaries, terminologies, unified units and etc. are the critical to accomplish the sematic interoperability. CDISC ADaM, SDTM, Define-XML and Controlled Terminology provide the support through the different ways.

The CDISC API is a good example. It transfers the implementation standards by the shared port, which makes the data exchange between CDISC and all the users easily accessible. Previously, different users may have different ways to maintain these PDF, XLSX, XML files, and need to use tools or manually track the change. Now, with the API, the latest information can be shared between the users and systems punctually and effectively.

Bringing these applications to the global, still has challenges. Currently, lots of clinical trials are conducted globally. If a site in east Asia is chosen, the multiple byte and language translation issues are inevitable. To make sure the information interoperability, we can't ensure the quality by translation company.

To conquer the multi-lingual challenge, CDISC developed the framework to support the standards being translated to each language on time. The standards and the terminology are the key to ensuring the meaning of exchanged data across different languages is preserved and unaltered.

Using China NMPA as example, currently, MedDRA, WhoDD has both English and Chinese version, LOINC and CDISC Core CTs Chinese version are translated by industry volunteers. If these terminologies accepted by the industry, then from the database builder procedure, we can implement these terms and make the multi-lingual package simultaneously. This will ensure the exchange data is accurate, and the overall data exchange process in clinical trials will be more efficient and consistent.

DATA EXCHANGE – MOVING FORWARD

With well-designed data structure and standardized terms, the next step is to achieve the process interoperability.

STREAMING DATA EXCHANGE

CDISC SDTM standards have been widely adopted in the industry already. Furthermore, a fair amount of data review applications has been developed based on them. Meanwhile, customers start to request the data to be delivered more frequently, say monthly or even biweekly. Under these circumstances, SDTM is treated as the structured raw data. Is it possible to submit SDTM datasets more frequently, like extract data from EDC? Nowadays, the video streaming service is very popular. It's constantly received by and presented to an end-user while being delivered by a provider (from Wikipedia). Is it feasible to submit the SDTM datasets in the way of streaming as a new method of data exchange?

A frame structure about the streaming service is illustrated below.

<TBD>
Considering most of the services are deployed in the cloud, turning the framework to the product should be less demanding. Internally, we are now using NodeJS to build it.

DYNAMIC PACKAGE REVIEW

As statistical programmers, a part of our job is to adjust the report format, add lines, remove blank space and make the wrap display more consistent. The tedious but necessary process can occupy 20% or more of our workload. With various formats of the report, such as RTF, PDF and HTML, it is common to lose information or display unexpected characters during the process of converting one to another.

These reports are often reviewed by statistician group or regulatory department. From their standpoint of view, reviewing a lab listing with thousand pages can be irritating. If the file is in RTF format, only OPENING it may take 1~5 minutes. Even after that, it's still frustrating to go through the entire report page by page.

Lots of programmers are developing the dynamic graph by using JMP Clinical, R/SHINY, Spotfire or Tableau. The interactive function makes the review more flexible and easier to gain insight. Also, it provides a way to gather more information on one screen, such as the patient profile graph.

Here is an example of the dynamic report prototype. (TBD, I will attach it later)

To make the dynamic style more consistent and popular, A clinical report standard to describe the metadata as SDTM or ADaM is required. Currently, Analysis Result Metadata (ARM) specification provides a set of elements (arm:AnalysisDatasets, arm:AnalysisResult and etc.) to describe the structure of a report. CDISC also initialed the REPORT XML a few years ago. We may extend these elements to cover more information on the reports, like column, header, footnote, source listing, etc.

By using a standardized structure, we may develop the dynamic report software or platform independently.

DATA SHARING BY BLOCKCHAIN

Data plays a crucial role nowadays, whereas it brings risks as well. When data is transferred across sites, SSL (Security Socket Layer) VPN, Cloud drivers (Box, Dropbox, OneDriver, etc.) are commonly used. These tools offer the secured transferring at one time point or within a short period ensured by users' credentials. As we provide the required information, we also lose the control of these transferred data. Is there a way to share the output information without transferring the dataset?

To increase the efficacy of clinical development, larger data pool is preferred for data mining and gaining knowledge about the specified cases. However, Data is considered as the property of each company. If only the summary analysis is shared, limited information can be provided. Whereas all the source data is shared, it brings the proprieties and maintenance issues. As the distributed and decentralized database, Blockchain provides a transparent and secured way to do so. Here are some preliminary ideas, (TBD)

CONCLUSION

Understanding the mechanism behind data exchange can be very helpful to follow the technical trends in our daily work. An effective data exchange is beneficial to all the related entities of a collaboration work.

REFERENCES

[1] A. Doan, A. Halevy, and Z. Ives. "Principles of data integration", Morgan Kaufmann, 2012 pp. 276

[2] Forms Data Format functions, PHP, 2003.
https://cct.me.ntut.edu.tw/ccteducation/chchting/aiahtm/computer/phphelp/ref.fdf.php.htm

[3] ASCII Encoding: Beginners, Newbies….Tom Ricken, 2019.
http://https//www.whoishostingthis.com/resources/asci/

*[4]Study Data Technical Conformance Guide V4.4, FDA*https://www.fda.gov/media/131872/download

[5] "Usage Survey of Character Encodings broken down by Ranking". w3techs.com. Retrieved 2019-11-11.

[6] Standards and Systems Interoperability: Definitions. 2017. file:///D:/Downloads/STDsInteroperability-Definitions-07-10-17FINAL.pdf

CONTACT INFORMATION

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

Author Name: Chao Wang
Company: FMD K&L
Address: 1300 Virginia Dr., Ste 408, Fort Washington, PA 19034
Email: chao.wang@klserv.com