Paper 171-2009

The Second CDISC Pilot Project
A Metastandard for Integrating Databases

Gregory Steffens, Principal Research Scientist, Eli Lilly & Co.
Ian Fleming, Senior Statistical Programmer Analyst, Genentech

Abstract

The first CDISC SDTM/ADaM pilot project created a test submission of one study to the FDA using CDISC data and metadata standards, in order to test that these standards meet FDA requirements (described in a paper Greg Steffens presented at SUGI 2007). The second CDISC pilot project objectives are to test 1.) the value of CDISC standards to create integrated databases (IDBs) and 2.) the new FDA safety review guidelines. This presentation focuses on the first objective and describes 1.) the collection and use of metadata in the second pilot and 2.) a very important drafted extension to the define.xml schema that supports row-level metadata and fills a gap in the metadata schema identified in the first pilot project.

Metadata for eight pilot project studies and three IDBs were populated in excel – excel was a user request as it has a more familiar interface than the SAS data set editors. SAS macros read this metadata and automated parts of the process when describing, building, comparing (studies to each other and the IDB specification) and validating the study ADaM and integrated databases. A SAS macro also automated the creation of the standard CDISC metadata; i.e. the define.xml file.

Row-level metadata enables the robust description of tall-thin data structures, such as exist in the SDTM data standard and that are becoming more common in clinical databases. The define.xml metadata, and the row-level extension, specify a standard set of database attributes that should be included in any data standard or database description - i.e. the metadata standard provides a standard language to describe data. This standard language, when it is made accessible to software in a standard structure, is essential to the automation of data flow from collection, through reporting, to integration, data mining, exchange of data between CROs and pharmaceuticals, and submissions. The CDISC define.xml standard may well turn out to have the most important impact that CDISC makes on the industry and to drug safety and reporting. Recently, some in the CDISC community are considering wiki methods of collecting data standards that should use the metadata standard in its technical infrastructure.

Metadata Structure and Row-Level Metadata

Metadata can be thought of as a list of database attributes put into a formal structure that can be accessed by software. It's immediately clear that a description of a database must include attributes like data set names, variable names and valid values of variables. When more thought is put into it, more attributes come to mind, such as data set labels, variable labels, SAS format associations, variable types (character, numeric, date, datetime, etc.), and so on. When designing metadata, this list of attributes is a core component. When designing database standards (like SDTM) and study database requirements, all the attributes in this standard list must be specified. Storing all this attribute information in a database – i.e. a metadatabase – allows software to access the information and automate what can only be done manually when this information is stored in unstructured formats such as word documents or pdf files.

The metadata standard used in the second CDISC pilot project consists of five components, implemented as excel spreadsheets and later converted to 4 SAS data sets and a SAS catalog of source entries. SAS macros then converted the SAS metadata into html and define.xml files for publication. The five components are:

Component	Description of Component
TABLES	Data set level attributes one row per data set.
COLUMNS	Variable level attributes one row per data set and

	variable.
COLUMNS_PARAM	Row level metadata, one row per data set, parameter variable, parameter value and parameter-related variable.
VALUES	Valid values and codelists associated with variables and parameter-related variables, one row per value list and valid value.
DESCRIPTIONS	A comment or derivation description that can be associated with a data set, variable and parameter-related variable.

The details of the five components can be found on the CDISC web page, for CDISC members, where I published them as part of the first pilot project (see the links from http://www.cdisc.org/membersonly/index.html). The columns_param component that stores row level metadata demonstrates the area where the define.xml schema was enhanced, so I will go into more detail about that component in this paper. Consider the vitals SDTM special observation class domain, as a simple example.

USUBJID	**VSTESTCD**	**VSLOC**	**VSORRES**	**VSORRESU**
1	SYSBP	STANDING	120	Mm mercury
1	HEIGHT			CM
1	WEIGHT			KG
1	BMI			KG/M**2

The important point to realize is that a simple description of a two-dimensional, rectangular data structure is an inadequate description of the VS domain. The VSLOC, VSORRES and VSORRESU variables have a different set of attribute values and the set of attribute values is dependent on the value of the VSTESTCD variable. What is required is a description of these three parameter-related variables (VSLOC, VSORRES and VSORRESU) for each value of the parameter variable (VSTESTCD). For example, VSLOC has a valid value of “STANDING” when VSTESTCD is “BPSYS”, but must be missing when VSTESTCD is “HEIGHT”, “WEIGHT” or “BMI”. The relationship between parameter variables and parameter-related variables, can also be seen with the VSORRES and VSORRESU variables. The range, format, and derivation descriptions of VSORRES are different for each value of VSTESTCD. The units stored in the VSORRESU parameter-related variable are different for each vital sign as well. Thus, the set of attribute values, that need to be specified for a parameter-related variable in a subset of rows that are defined by the parameter variable value, is identical to the set of attributes that need to be specified for a variable. The columns and columns_param metadata sets contain the same set of attributes and differ only by additional primary keys in columns_param that identify the parameter values. The decision to store data in short-wide or tall-thin data set structures has no affect on the amount of metadata definition that is required, although it initially seems to some people that tall-thin data structures require less definition that short-wide structures. The following short-wide data set requires an almost identical amount of definition as the above tall-thin data set. In a sense, when defining tall-thin data sets you need to include a definition of virtual variables – i.e. variables that would exist if the data were stored in a short-wide structure.

USUBJID	SYSBP	BPSYSLOC	BPSYSU	HEIGHT	HEIGHTU	WEIGHT	WEIGHTU	BMI	BMIU
1	120	STANDING			CM		KG		

The Draft Extension of the Define.xml File to support Row Level Metadata

The schema of the define.xml file was extended by a joint effort between the pilot team and the CDISC ODM team. The extension supports the row level metadata that was collected in the columns_param metadata component used in the pilot project. An xml tag, named “ValueList” was modified to allow the definition of the subset of rows that an ItemDef can be applied to. That is, the ValueList tag allows the definition of the parameter variable values and the attachment of an ItemDef to an item for the subset of rows defined by the parameter variable. The concept of an “Item” thus becomes more refined than that of a variable in a SAS data set or a column in a relational database, as it takes on the concept of a parameter-related variable. We may store data in 2-dimensional relational tables, but metadata adds a critical third dimension to the description and use of the data.

Collection and Use of Metadata in the CDISC Pilot Project

The second CDISC pilot project uses pediatric data from eight studies for three different compounds. The first compound contains four studies worth of data, while the second and third compounds contain two studies worth of data each. In addition to the individual studies, integrated analyses will be performed for each compound, which necessitates the creation of compound level databases. Each of the study and compound level databases is actually comprised of two smaller databases, one for SDTM data and one for ADaM data. The entire submission contained a total of 19 unique databases, each requiring their own metadata and related submission document compilation.

Excel spreadsheets were used to capture the individual domain metadata for each study and integrated database. This spreadsheet contains unique tabs to store information about the individual CDISC versions being used, domain information, the columns of the domain, any row level metadata, and terminology or formats utilized in that domain as described earlier. The metadata captured in this Excel spreadsheet contains all information necessary, as specified, in the SDTM and ADaM implementation guides to produce valid and usable define.xml documentation to be included in the final submission to the FDA. The unique process and team dynamics presented challenges that required that a number of specialty tools be developed in order to compile, process, and check the metadata.

The first tool, a SAS macro named mdcompile, compiled all of the separate domain Excel spreadsheets for each database into a master metadata Excel spreadsheet. Due to the large number of metadatabases that needed to be processed through this macro, the macro was designed to dynamically adjust to the unique attributes of each metadatabase requiring no macro parameters upon calling the macro. Each individual Excel spreadsheet was compiled by a different team member to speed up the production of these deliverables and the individual databases. This required that the macro be extremely robust in it's handling of inconsistencies in the entered metadata. For any inconsistencies that could not be handled programmatically, reports are generated to help identify issues quickly so that the proper team member could be notified and corrections could be made. Some basic checking to assure consistency between domains and also input of some CDISC column attributes were handled by the macro to help reduce the amount of user maintained metadata. These spreadsheets were compiled prior to programming of the domains. This allowed the individual programmers to get an idea of the data structures and programming intricacies prior to the start of programming of the domains.

During the programming of the domains, a SAS macro named mdattribs was used to populate the SAS dataset attributes based on information from the SDTM and ADaM implementation guides. This macro was called at the end of each domain generating program and reads metadata content stored and maintained in a central, secure location to generate SAS code that defines the data structure, including data set names, variable names, labels, formats, etc. This process structure allows this metadata to be maintained in one place and then pushed to the produced domains with the invocation of the macro. This allows the programmers of the domains to concentrate on the domain content without having to spend time producing and maintaining standard objects and information.

Once the domain data sets were programmed and populated, there are two additional SAS macros that were developed to ensure compliance of the compiled metadata to standard metadata templates and compliance between the compiled metadata and the produced data sets. The mdcompare SAS macro uses a centrally maintained representation of the metadata spreadsheet based on the specifications laid out in the SDTM and ADaM implementation guides in order to compare against the individual compiled metadata spreadsheets from the individual study or integrated metadatabases. This ensures that the metadatabases compiled by the pilot team conform to the standards laid out in the SDTM and ADaM implementation guides. In addition, the mdcheck SAS macro checks the produced SAS datasets against the compiled metadata to ensure that the those two objects are consistent for every domain produced in each metadatabase and database.

Together, these tools accomplish the goals of simplifying the complex process of producing the submission deliverables with very challenging human resource dynamics, while simultaneously ensuring that the deliverables are of the highest quality and conform to all CDISC standards.

Conclusion - Uses of Metadata Standard for the Industry

Metadata should be an integral part of the data flow and not an afterthought. Populating standard metadata prescriptively at the start of a study, rather than descriptively at the end of a study, enables automation, easier and better regulatory submissions, and the integration of multiple study databases into one IDB for analysis that could not be possible at the level of individual study analyses.

The storage of data standards and study data specifications in one standard metadatabase structure, leads to very significant improvements in technology and process. Vendors are using the ODM and define metadata standards to improve study setup by integrating the definitions of source data sets and eCRFs. Storing study analysis data set and integrated database requirements in metadata leads to greatly improved automation in the description, creation, validation, submission and data mining of the vast amount of clinical and non-clinical data. Being able to do meta-analysis of integrated study databases with more ease that has existed will improve the ability to analyze safety and efficacy of drugs. The FDA can even integrate data across different companies to look for safety signals that might not appear in individual studies or IDBs.

The communication of data requirements between a pharmaceutical and a CRO (or other outsourcing agency) can now be standardized. If we all use the same metadata "language" to describe data specifications to each other, communication of the huge about of detail will be greatly improved and made available to software access as well as human access. As vendors, such as Formedix, develop applications that populate standard metadata that contain data specifications that can be interoperable between different vendors, a new and better way of managing data will evolve that makes data easier to collect, clean and analyze. Adoption of standard metadata can also facilitate the creation of new industry standard data structures, like the SDTM. Individual companies could submit proposed data standards to the standards setting organization using a common metadata language, that ensures the proposal is complete and unambiguous. This could lead to a new way to define data standards where we can leverage more resources in a wiki – like environment, rather than create small teams of people who try to define an industry standard that is drafted and put out for review.

Contact Information

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

Name:	Greg Steffens
Enterprise:	Eli Lilly & Co.
Address:	Lilly Corporate Center
City, State ZIP:	Indianapolis, IN
Work Phone:	317-651-4857
Fax:	317-277-7839
E-mail:	steffensgc@lilly.com

Name:	Ian Fleming
Enterprise:	Genentech
Address:	1 DNA Way
City, State ZIP:	South San Francisco, CA
Work Phone:	650-467-5038
Fax:	650-225-7981
E-mail:	fleming.ian@gene.com