

The Use of Metadata in Creating, Transforming and Transporting Clinical Data

Gregory Steffens
Director, Data Management and SAS Programming
ICON Development Solutions

- Data flow is comprised of transformation and derivation
 - Transformation – restructures, re-attributes and integrates
 - Derivation – creates new values in a target database that do not exist in the source

- Data transformations comprise a large portion of what we do
 - Extract from data collection/cleaning applications to a standard
 - Integrate ancillary data like lab, randomization, QOL, etc.
 - Creation of analysis database
 - Integrate multiple studies into an integrated database

- Therefore, there is a need to define, create, validate and describe databases efficiently and with a consistently high level of quality
- A combination of reusable code driven by standardized metadata is the answer – the Data Transformation Engine (DTE)
- The code must not make any assumptions of the source and target databases, this is communicated to the code via metadata
- The metadata structures must be absolutely standardized across all applications and data standards
- Metadata is populated prescriptively and drives the data flow

- A standard list of attributes to include in any description of a database or of a data standard
- Put in a standard set of data structures that can be read by the DTE code
- The attributes must be highly structured in order to be usable by DTE code

- **Data Set Level**
 - Short/long names, data set location, order
- **Variable level**
 - Short/long name, type, length label, primary key flag, format, value list name, suppqual flag, code/decode relationship, order,
- **Valid values**
 - Value list name, start/end value, short decode, long decode
- **Descriptions**
 - Source name, derivation description
- **Row-level attributes**
 - Identical to variable level attributes but for subsets of rows defined by a parameter variable value

- Necessary to fully describe tall-thin data set structures

USUBJID	SYSBP	BPSYSLOC	BPSYSU	HEIGHT	HEIGHTU	WEIGHT	WEIGHTU	BMI	BMIU
1	120	STANDING	mm Mg	185	CM	90	KG	26.3	Kg/m**2

USUBJID	VSTESTCD	VSLOC	VSORRES	VSORRESU
1	SYSBP	STANDING	120	Mm Mg
1	HEIGHT		185	CM
1	WEIGHT		90	KG
1	BMI		26.3	Kg/M**2

- Structured content to enable programmatic access
- Storage structure is separate from publication structure – maximize programmatic access and user friendly access by people
- Also separate from entry format
- Maximize sharing of information within the metadata, e.g. values lists and descriptions

- Rigorously standardized for all database and standard descriptions, no metadata design change for database types!
- Maximize structured information and programmatic access, e.g. primary keys flagged instead of listed
- Enter once; use many. e.g. descriptions and values
- Derivation logic in descriptions though

- It is critical to explicitly define the objectives. Many disagreements arise from an unstated difference in assumed objectives
- Objectives allow evaluation of the success of the metadata design; e.g. retrospective description for esubmission or prescriptive enabler of automation.

- Single metadata design for everything
- Minimize duplicate entry
- Include enough attributes to enable automation
- Separate metadata design from publication and from entry
- Generalize SAS macros, don't specialize for any specific database standard

- Creating study data requirements from data standards
- Publish data requirements in several formats – pdf, xml, html, etc.
- Publish data requirements with different levels of detail for different user types
- Implement all database attributes from the requirements with one macro call
- Create version 5 transport files with renames and length changes of names and labels
- Create integrated databases
- Create all decode variables with one macro call

- Move variables in and out of suppqual domains
- Validate that a database follows a standard and a study requirement
- Compare a new data requirement to a standard or other studies
- Create 0-obs data sets
- Automate TFL shells
- Transform databases from any structure to any other structure ... but what else do we need to do that?

- So far, we have metadata that describes databases
- The next step is standard map metadata to define transformation types from one database to another
- Map metadata associates one observation of source metadata with one or more observations of target metadata
- This mapping information is read by SAS macros that generate all the SAS code to create a target database from a source database
- Map metadata should be separate but integrated with metadata

- **%dtmap**
 - (source_mdlib=m,source_prefix=raw_,target_mdlib=m,target_prefix=target_,maplib=me,inlib=raw,outlib=sdtm,suppqual_make=yes)
- **%mdprint**
 - (mdlib=m,inprefix=target_,html=yes,htmlfile=c:\metadata\target_define.html)
- **%md2odm**
 - (mdlib=m,outxml=c:\metadata\SDTM.xml,StudyName=ICR_STDM_Standard 3.1.2,ProtocolName=SDTM,StandardName=SDTM 3.1.2,StudyDescription=SDTM Standard,defineVersion=1.0.0,odmversion=1.2,crt_prefix=def)

- Standards are a means to automation and automation is a means to efficiency and quality
- Much less code invented and validated for individual studies
- Change in skill set requirements for data flow – separate coding skills from database and clinical skills
- More productivity from junior staff
- Faster, better and cheaper.