THIRD-PARTY DATA TRANSFER EXAMPLE OF METAPROGRAMMING

Greg Steffens

Associate Director

Gilead Sciences

BASAS November 3, 2016

METAPROGRAMMING DEFINED

SINGLE-USE CODE COPIED FROM PROJECT TO PROJECT

- Write a program in one project and copy it to the next project
- Requires modification in the new projects to handle new data situations and requirements
- Advantages
  - Easy to do – the code is simpler to write
  - If you have enough highly experienced programmers, this is effective
- Single-use code is code that is validated for a single project use. Validation of this type does not guarantee it will work for another project.

SINGLE-USE CODE COPYING DISADVANTGES

- You need people who understand when and how to make modifications in new projects, so as you have more projects you need more programmers with high experience levels. Therefore, this is not a scalable solution.
- There is an increasing level of risk of error in the modifications
- The code becomes "patchy" spaghetti code that is increasingly difficult to understand, the programmers end up not really knowing what the code does
- You end up with many versions of the original code with different subsets of modifications. The propagation of the code will not be linear.
- You are constantly re-validating the same code

WHAT IS META-PROGRAMMING?

- Meta-programming provides code that can be used across a very wide scope of projects and processes without modification, maximizing code reuse.
- Meta-programming includes
  - Code that contains much more decision-making logic
  - Code that is configurable by metadata and macro parameters, rather than manual changes to code
  - Standards that reside in metadata to provide metadata content
  - Multi-use code validation that reduces project-level validation
- Meta-programming provides highly effective and flexible automation
- Greatly increases transparency about how data are created
- Code that generates code

WHAT IS IT, TO BE A PROGRAMMER?

- The further you progress through these evolutionary stages, the more benefit you will see.
- You are following best programming practices by extending the functionality of SAS. Think of it as PROC writing. The knowledge experience you put in meta-programming will greatly increase the scope of the value you deliver.
- There are different types of programming roles
  - Data analyst who uses programming the minimum amount to accomplish the analysis
  - Full-time programmers who work study to study using a mix of single-use and multi-use code
  - Systems programmers who develop highly re-useable code that the programmers in the other roles call.

EXAMPLE

THIRD-PARTY DATA EXCHANGE PROCESS

1. Create data transfer standard in metadata
    1. Pharma and third-party data source collaboration (TPDS)
    2. Could be TPDS specific or shared across TPDS types
2. For each data exchange with the lab or other third-party data source create data specifications in metadata, making study changes if necessary – the Data Transfer Specification (DTS) – from the standards metadata or other study metadata
3. Publish the data specification from the metadata in human-friendly format – e.g. define file
4. Create the data according to the specification
    1. Third-party data source
5. Verify that the data conforms to the specification
    1. Pharma does so on receipt of the data
    2. Collect metrics on data quality from third-party data source
6. Convert data to internal standard if necessary
    1. Some third-party sources can’t give us data in the form we need

TPDS PROCESS

Step	Responsible	Metaprogram Module
Create data standard in metadata	Pharma and third-party data source (TPDS)	mdmake, mdbuild
Create study data specification in metadata	Pharma or TPDS	mdmake, mdbuild, mdcompare
Publish data specification in human-friendly format- e.g. define file	Pharma or TPDS	md2html md2odm
Create the data according to the specification	TPDS	dte, mdatribs, single-use programs, etc.
Verify that the data conforms to the specification	Pharma	mdcheck
Convert data to standard format	Pharma	dte

WHAT DOES INTERNAL SDTM CREATION LOOK LIKE?

Step	Responsible	Metaprogram Module
Create data standard in metadata	Pharma	mdmake, mdbuild
Create study data specification in metadata	Pharma	mdmake, mdbuild, mdcompare
Publish data specification in human-friendly format- e.g. define file	Pharma	md2html md2odm
Create the data according to the specification	Pharma	dte, mdatribs, single-use programs, etc.
Verify that the data conforms to the specification	Pharma	mdcheck
Convert data to standard format	Pharma	dte

WHAT DOES OUTSOURCED SDTM CREATION LOOK LIKE?

Step	Responsible	Metaprogram Module
Create data standard in metadata	Pharma	mdmake, mdbuild
Create study data specification in metadata	Pharma and CRO	mdmake, mdbuild, mdcompare
Publish data specification in human-friendly format- e.g. define file	Pharma	md2html md2odm
Create the data according to the specification	CRO	dte, mdatribs, single-use programs, etc.
Verify that the data conforms to the specification	Pharma	mdcheck
Convert data to standard format	Pharma	dte

WHAT DOES OUTSOURCED ADAM CREATION LOOK LIKE?

Step	Responsible	Metaprogram Module
Create data standard in metadata	Pharma	mdmake, mdbuild
Create study data specification in metadata	Pharma and CRO	mdmake, mdbuild, mdcompare
Publish data specification in human-friendly format- e.g. define file	Pharma	md2html md2odm
Create the data according to the specification	CRO	dte, mdatribs, single-use programs, etc.
Verify that the data conforms to the specification	Pharma	mdcheck
Convert data to standard format	Pharma	dte

EXAMPLES OF METAPROGRAMMING MODULES (SAS MACROS)

mdmake	Creates metadata from other metadata – e.g. creates study metadata from data standards metadata
mdbuild	Builds metadata to describe an existing data library
Mdcompare / mdcompare_print	Compares metadatabases to each other, such as a study requirement to a standard or a study to a study
Md2html/md2odm	Publish in define html or xml format
dte	Top level macro that users call to transform data from one format to another, e.g. raw to SDTM to ADaM to IDB
Mdatribs	Apply attributes defined in metadata to a data library
mdcheck	Checks data and reports discrepancies with the metadata
mdfreqvals	Creates the values metadata set (supplements mdbuild)

ALL DATA FLOWS IN YOUR COMPANY HAVE A COMMON META-PATTERN

- Find the underlying patterns
  - we're just creating one relational database from another, it's mere content that changes across processes
- Use meta-programming methods to automate these patterns
- Improvements
  - Single technology for all these processes, instead of manual single-use programming and different systems developed for each process
  - Lesser study-level software validation because less code is created at the study level when there is more metaprogram modules available
    - Who uses the PUT statement for TFLs today, rather the proc report?

IS INDUSTRY METAPROGRAMMING POSSIBLE?

- What if we all shared the standard metadata design?!
  - Exchange of data specifications between pharma, TPDS, CROs, regulatory agencies would all use the same metadata design
- This would enable shared meta-program modules
  - To create define files
  - To create data
  - To check data to specifications
  - To … lots of things
- Not reinventing that wheel over and over again
  - Regulatory agencies get the same thing from all companies
  - Your moves from TA to TA and company to company become easier

IMPLEMENTATION CONSIDERATIONS

SILOS

- Silos of Process - can't see the underlying patterns under each process, resulting in silo-ed thinking and solutions
  - This is DTS but that is SDTM so the solution must be different
- Silos of Organizations
  - One group does DTS another group does SDTM, so the technology solution must be different
- Solution ideas
  - Develop technical solutions with the big picture in mind
  - Advocate for cross-silo solutions
    - It will be seen as not possible or overly complex, but develop the technology to reduce assumptions about process so that it will become evident, over time, that the technology works everywhere

FEAR

- Fear of code
  - Roles are very different
  - Systems development – full time creation of meta-programming and metadata design
  - Study programming – many study programmers think metaprogramming is complicated and scary
- Fear of change
  - We’ve always done it the old way and we change only when it’s a crisis
- Fear of success
  - What will happen to my job if this automation really works?
- Fear of failure - past experience that is negative
  - Someone failed in the past so no one can succeed now and in the future

OTHER

- Future expectations from “vaporware” promises
  - Software that isn’t written yet can be described to do anything and everything
  - Promises are much easier than delivery
- Perfection disabling improvement
  - Someone can think of an example that happens 1 in a 1000 studies that may not work so don’t make any improvement for the 999 studies
  - Be careful not to exclude examples that should be included in requirements

WANT MORE INFORMATION?

- PhUSE SDE scheduled February 23, ,2017 will cover these kinds of topics
- The location is on the Gilead campus