



Meta-programming in SAS Clinical Data Integration : a programmer's perspective

Mark Lambrecht, PhD
Phuse Single Day Event
Brussels , February 23rd 2010.

Copyright © 2010, SAS Institute Inc. All rights reserved.

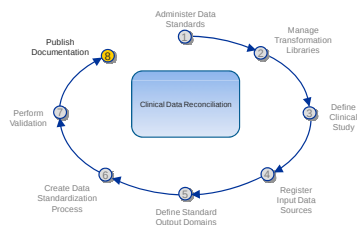


Contents

- SAS Clinical Data Integration : an introduction
- Meta-programming in SAS Clinical Data Integration
 - Use case 1 : mapping transformation
 - Use case 2 : coding transformation
 - Use case 3 : SDTM validation
 - Use case 4 : publish define.xml
- SAS language interfaces to metadata and their use in clinical data standards generation

Copyright © 2010, SAS Institute Inc. All rights reserved.

SAS Clinical Data Integration



Clinical Data Integration is the SAS solution for converting captured, legacy and external clinical data into tabulation and analysis datasets using one unique version of the metadata.

It delivers integration capabilities with CDISC ODM and EDC systems, and with SAS Drug Development

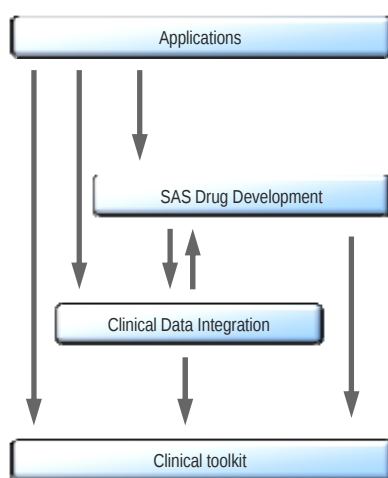
It delivers out-of-the-box CDISC standard, protocol and submission elements in a hierarchical and secured environment.

Data quality checks are provided by the Clinical Standards Toolkit as a customizable module.

It manages single step conversion from external to SDTM data model

Copyright © 2010, SAS Institute Inc. All rights reserved.

Clinical Standards Toolkit – Context



Copyright © 2010, SAS Institute Inc. All rights reserved.

SAS Clinical Data Integration Functional Components

CDISC and company standards	• Integrated data standard models and associated CDISC metadata
Clinical data flow capability	• Integration with SAS Drug Development
Fast access to source and external data	• Integration scenario with EDC, operational systems and lab files
Clinical project and study management	• Project and studies definition
Visual data transformation	• Data mapping
Generated SQL and SAS code	• Traceability of conversions at level of code
Integrated metadata and impact analysis	• Define.xml and compliance adherence is result of integration, not extra process
Data quality	• Validation against controlled terminology and solution intelligence guarding over mapping objective
User-written SAS code integration	• Specialized code (derived flag derivation) – complex tasks for SAS programmers
Collaboration	• Increased efficiency between clinical data managers, programmers and project coordinators

Copyright © 2010, SAS Institute Inc. All rights reserved.

SAS Clinical Data Integration Workflow

- Import Data Standards
- Customizing Data Standards
- Configuration of Defaults
- Create a Clinical Component
- Define Domains
- Utilize Metadata in transformation process
- Monitor Development Status
- Analyze Use of Data Standard

Copyright © 2010, SAS Institute Inc. All rights reserved.

Importing Data Standards

- CDISC Industry Standards shipped with SAS Clinical Data Standards Toolkit
- Adaptable to custom implementations
 - SDTM +/-
 - Internal Standards

Copyright © 2010, SAS Institute Inc. All rights reserved.

Data Standard Loaded in Metadata

SDTM Classes

SDTM Domains

Copyright © 2010, SAS Institute Inc. All rights reserved.

Customizing Data Standards

- SDTM Standard LB Domain

#	Name	Description	Type	Length
27	LBSPEC	Specimen Type	Character	
28	LBSPCOND	Specimen Condition	Character	
29	LBMETHOD	Method of Test or Examination	Character	
30	LBBLFL	Baseline Flag	Character	
31	LBORVFL	Derived Flag	Character	
32	LBPAST	Posting Status	Character	
33	LBTOX	Toxicity	Character	
34	LBTOXGL	Standard Toxicity Grade	Character	
35	VISIT	Visit Name	Character	
36	YVISITNUM	Visit Number	Numeric	
37	YVISITDY	Planned Study Day of Visit	Numeric	
38	LBOTC	Date/Time of Specimen Collection	Character	
39	LBENDOTC	End Date/Time of Specimen Collection	Character	
40	LBOT	Study Day of Specimen Collection	Numeric	
41	LBPT	Planned Time Point Name	Character	
42	LBPTNUM	Planned Time Point Number	Numeric	
43	LBELTM	Elapsed Time from Reference Point	Character	
44	LBPTREF	Time Point Reference	Character	

Copyright © 2010, SAS Institute Inc. All rights reserved.

Customizing Data Standards

- Customized LB Domain (SDTM +/-)

#	Name	Description	Type	Length	Format	Form
1	STUDYID	Study Identifier	Character	40 (None)	(None)	
2	DOMAIN	Domain Abbreviation	Character	8 (None)	(None)	
3	USUBID	Unique Subject Identifier	Character	40 (None)	(None)	
4	LBSEQ	Sequence Number	Numeric	8 (None)	(None)	
5	LBTESTCD	LAB Test or Examination Short Name	Character	8 (None)	(None)	
6	LBTEST	LAB Test or Examination Name	Character	40 (None)	(None)	
7	LBORRES	Result or Finding in Original Units	Character	25 (None)	(None)	
8	LBORRESU	Original Units	Character	40 (None)	(None)	
9	LBSTRESN	Numeric Result/Finding in Standard Units	Numeric	8 (None)	(None)	
10	LBSTNLO	Reference Range Lower Limit-Std Units	Numeric	8 (None)	(None)	
11	LBSTNHI	Reference Range Upper Limit-Std Units	Numeric	8 (None)	(None)	
12	LBORCM	Unit Code	Character	200 (None)	(None)	
13	LB3DVGM	3 x Normal Flag	Character	1 (None)	(None)	
14	LBBLFL	Baseline Flag	Character	1 (None)	(None)	
15	LBORVFL	Derived Flag	Character	1 (None)	(None)	
16	VISIT	Visit Name	Character	40 (None)	(None)	
17	YVISITNUM	Visit Number	Numeric	8 (None)	(None)	
18	LBOTC	Date/Time of Specimen Collection	Character	64 (None)	(None)	
19	LBPTNUM	Planned Time Point Number	Numeric	8 (None)	(None)	

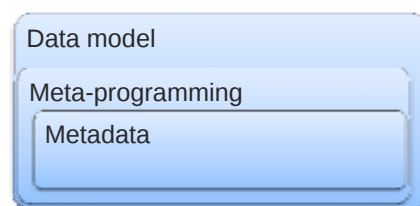
Copyright © 2010, SAS Institute Inc. All rights reserved.

Metadata ?

- Variable-level, column-level metadata
- Additional CDISC metadata : code lists, terminology, computational derivations
- Macro metadata / parameters
- Technical metadata : servers, users, architecture
- Macro programming is really a long chain of inputs and outputs with interfaces definitions

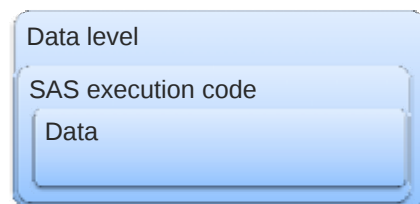
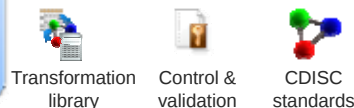
Copyright © 2010, SAS Institute Inc. All rights reserved.

Data model programming or “meta-programming”



Meta-programming

Wilcock and Potula, PhuUSE2009 – paper AD10.



Physical data



Copyright © 2010, SAS Institute Inc. All rights reserved.

Data model programming or "meta-programming"

Data model

article discussion edit this page history

Metaprogramming

From Wikipedia, the free encyclopedia

How do I do it? What's the programming language? For the management language, see Metaprogramming (management).

Metaprogramming is the writing of computer programs that write or manipulate other programs (or themselves) as their data, or that do part of the work at compile time that would otherwise be done at runtime. In many cases, this allows programmers to get more done with less code than if all functions they would like to make all the code manually, and gives programs greater flexibility to efficiently handle new situations without recompilation.

The language in which the metaprogram is written is called the *metalinguage*. The language of the programs that are manipulated is called the *object language*. The ability of a programming language to do these things is called *reflection* or *reflexivity*.

Reflection is a valuable language feature for building metaprogramming. Having the programming language itself as a *first-class data type* (as in *Lisp*, *Fort* or *Perl*) is also very useful. *Generic programming* involves a metaprogramming facility within a language, in those languages supporting it.

Metaprogramming usually works through one of two ways. The first way is to expose the internals of the run-time engine to the programming code through *application programming interfaces* (APIs). The second approach is dynamic evaluation of string expressions that contain programming commands. Thus, "programs run with programs". Although both approaches can be used in the same language, most languages choose to use one or the other.

navigation

- Main page
- Contents
- Featured content
- Current events
- Random article

to search

Go Search

list of all pages

- About Wikipedia
- Community portal
- Help and guides
- Recent changes

Copyright © 2010, SAS Institute Inc. All rights reserved.

Repositioning of roles in clinical data integration

Role 1 : data analyst

Base SAS programmer that can transform raw data into SDTM data and ADaM data.

At the same time of translating the data models into executable code, the data analyst is working together with the...

Role 2 : data modeller

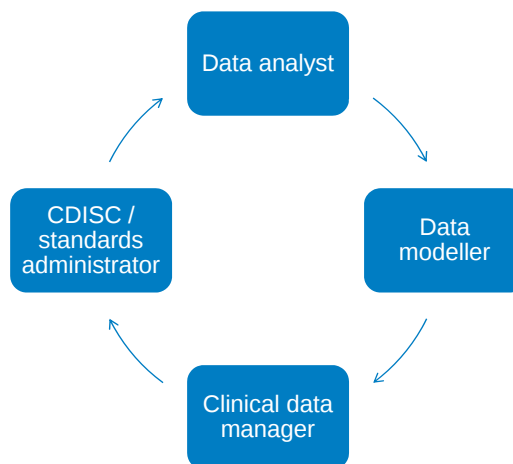
The data modeller is involved in data model programming and validation of CDISC data

Role 3 : clinical data manager

Assures that clinical data is of good quality and analysis-ready

Role 4 : CDISC administrator

Administers the clinical standards, versions, studies and monitors quality of submission data





Define.xml

Dataset	Description	Structure	Purpose	Keys	Location
AE	Adverse Events	Events - One record per adverse event per subject	Tabulation	STUDYID, USUBJID, AETERM, AESTDTC	ae.xml
CM	Concomitant Medications	Interventions - One record per medication intervention episode per subject	Tabulation	STUDYID, USUBJID, CMTRT, CMSTDTC	cm.xml
DM	Demographics	Special Purpose - One record per subject	Tabulation	STUDYID, USUBJID	dm.xml
DS	Disposition	Events - One record per disposition status or protocol milestone per subject	Tabulation	STUDYID, USUBJID, DSSTDTC	ds.xml
EX	Exposure	Interventions - One record per constant dosing interval per subject	Tabulation	STUDYID, USUBJID, EXTRT, EXSTDTC	ex.xml
LB	Laboratory Tests	Findings - One record per lab test	Tabulation	STUDYID, USUBJID, LBTESTCD, LBTESTNM, LBTESTRES, LBTESTU, LBTESTDT	lb.xml

Copyright © 2010, SAS Institute Inc. All rights reserved.

What is the SAS Metadata Model ?

- Object-oriented, hierarchical model
 - Objects and classes
 - Associations between classes
 - Inheritance of attributes and associations
 - Subclassing to extend behaviours
- SAS Clinical Data Integration has extended the SAS Metadata Model with new Metadata Types
 - Studies
 - Submissions
 - Clinical Domains

Properties:

Name	Value
Classification	Special Purpose
DomainIdentifier	
DomainPurpose	DM
DomainStructure	Tabulation
IsCustom	One record per subject
IsSplit	
IsSupplemental	
UsesTemplate	true

SAS language interface to metadata

- 3 methods

SAS Java Metadata Interface, the

SAS Microsoft .NET Framework Application
Services Interface, and the

Base SAS language interfaces for external
customers

Copyright © 2010, SAS Institute Inc. All rights reserved.

Base SAS language interface to metadata

- Procedures : metadata procedure
- Data step interface

Copyright © 2010, SAS Institute Inc. All rights reserved.

Base SAS language interface to metadata

- Procedures : metadata procedure

Example :

```
proc metadata
    in='<GetMetadata>
        <Metadata>
            <PhysicalTable Id="A58LN5R2.AR000001"/>
        </Metadata>
        <Ns>SAS</Ns>
        <Flags>1</Flags>
        <Options/>
    </GetMetadata>';
run;
```

Copyright © 2010, SAS Institute Inc. All rights reserved.

Base SAS language interface to metadata

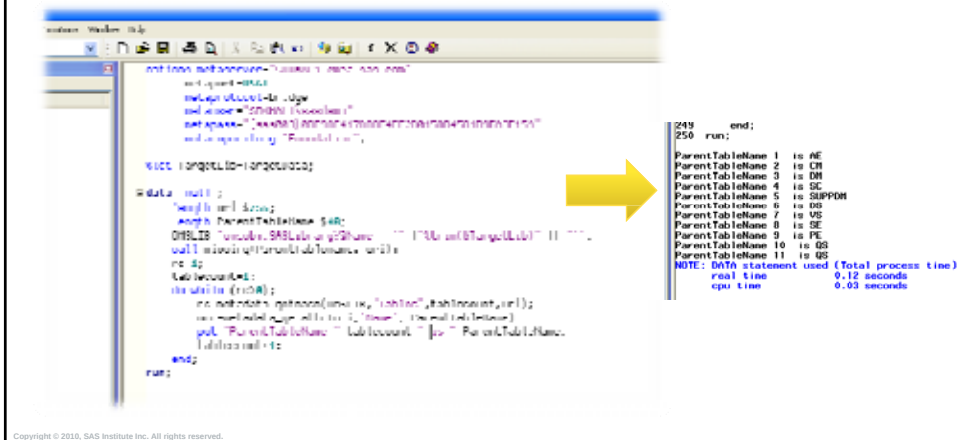
- Data step interface functions (not exhaustive)

Function Syntax	Description
METADATA_RESOLVE(uri, type, id)	Resolves a metadata URI into a specific object type
METADATA_GETATTR(uri, attr, value)	Returns the named attribute for the object specified by the URI
METADATA_GETNASL(uri, n, asn)	Returns the nth named association for the object URI
METADATA_GETNASN(uri, asn, n, nuri)	Returns the nth associated object of the association specified
METADATA_GETNATR(uri, n, attr, value)	Returns the nth attribute on the object specified by the URI
METADATA_GETNOBJ(uri, n, nuri)	Returns the nth object matching the specified URI
METADATA_GETNPRP(uri, n, prop, value)	Returns the nth property on the object specified by the input URI
METADATA_GETNTYP(n, type)	Returns the nth object type on the server
METADATA_GETPROP(uri, prop, value)	Returns the named property for the object specified by the input URI

Copyright © 2010, SAS Institute Inc. All rights reserved.

Base SAS language interface to metadata

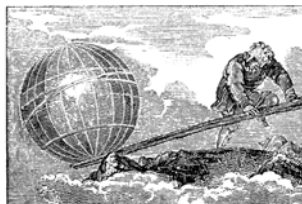
- Data step interface functions



Meta-programming in SAS Clinical Data Integration

- Delivers standardized code, added-value programming
- Robust validation routines
- Collaboration between different roles in the generation of submission-quality data : the programmer, data modeller/mapper and the clinical data manager.
- Avoidance of creation of huge unmanageable macro libraries and increase of re-usability

*GIVE ME A PLACE TO STAND AND I WILL
MOVE THE EARTH (Archimedes)*



The engraving is from *Mechanic's Magazine*
([cover of bound Volume II](#), Knight & Lacey, London, 1824) Courtesy of the [Annenberg Rare Book & Manuscript Library](#), University of Pennsylvania Philadelphia, USA



Mark Lambrecht, PhD
SAS
Hertenbergstraat 6
B-3080 Tervuren, Belgium.

Work Phone: +32 2 766 07 53
E-mail: mark.lambrecht@sas.com

Copyright © 2009, SAS Institute Inc. All rights reserved.