



Conversion of Clinical Data Across Data Standards

This paper presents a metadata-based methodology for converting clinical data across data standards using an in-house application with even SAS8.

Author: Hitesh Sharma

Conversion of Clinical Data Across Data Standards Using In-House Application

Hitesh Sharma

Abstract:

Various business needs are driving the task of clinical data integration in Pharmaceutical industry. The enterprises using SAS8 databases, can accomplish data integration by in-house application, working on metadata based transformations. It offers great cost and quality benefits.

Introduction:

The healthcare industry has seen a significant shift on emphasis on data standards. The Pharmaceutical companies have been defining their own data standards for some time now. Unfortunately, there have been variations in interpretation and compliance to those data standards. In addition, the data standards have changed within organizations because of internal and external factors. The business need to integrate data across protocols have resulted in evolution of interchange standards like those by Clinical Data Interchange Standard Consortium (CDISC). The companies also have redefined their standards to have easy portability to or compatibility with such interchange standards. Along with redefining of data standards, comes the task of bringing the existing data to the new data standard (Ref. Fig 1). Doing these transformations have been a considerable task for the companies.

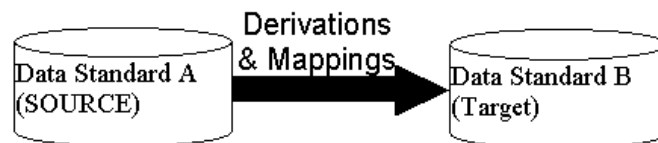


Figure 1: Standard Transformation

Since the clinical data is usually in SAS format, it would not be profitable to use a non-SAS platform to do such transformations. In Pharmaceutical industry clinical data is mostly used in SAS format.

Traditionally, people have been doing such transformations by writing lengthy SAS code for each integration transformations needed. As new protocols get added to the transformation, the programs become too complex and difficult to maintain. Also, the validation costs for such codes are huge.

With the advent of Business Intelligence (BI) infrastructure by SAS in version 9, we can see that above mentioned transformations have quite a different and convenient approach. The cost involved in implementing the transformations in SAS9 goes high and may not be profitable, unless the organization have committed to using the SAS9 products for a wide use.

Framework for Using Metadata Based Approach:

We recommend that the transformations between data standards should use metadata-based approach. This enhances reusability of the transformation logics and automates the process of mapping the source metadata with the target metadata using standard transformations. It would also efficiently bypass the manual coding.. This would result in high quality and lower costs.

The proposed framework (Ref. Figure 2) provides the capability to design and use metadata for the source, target and transformations. The following diagram gives a broad picture of the proposed framework. The description of the components follows:

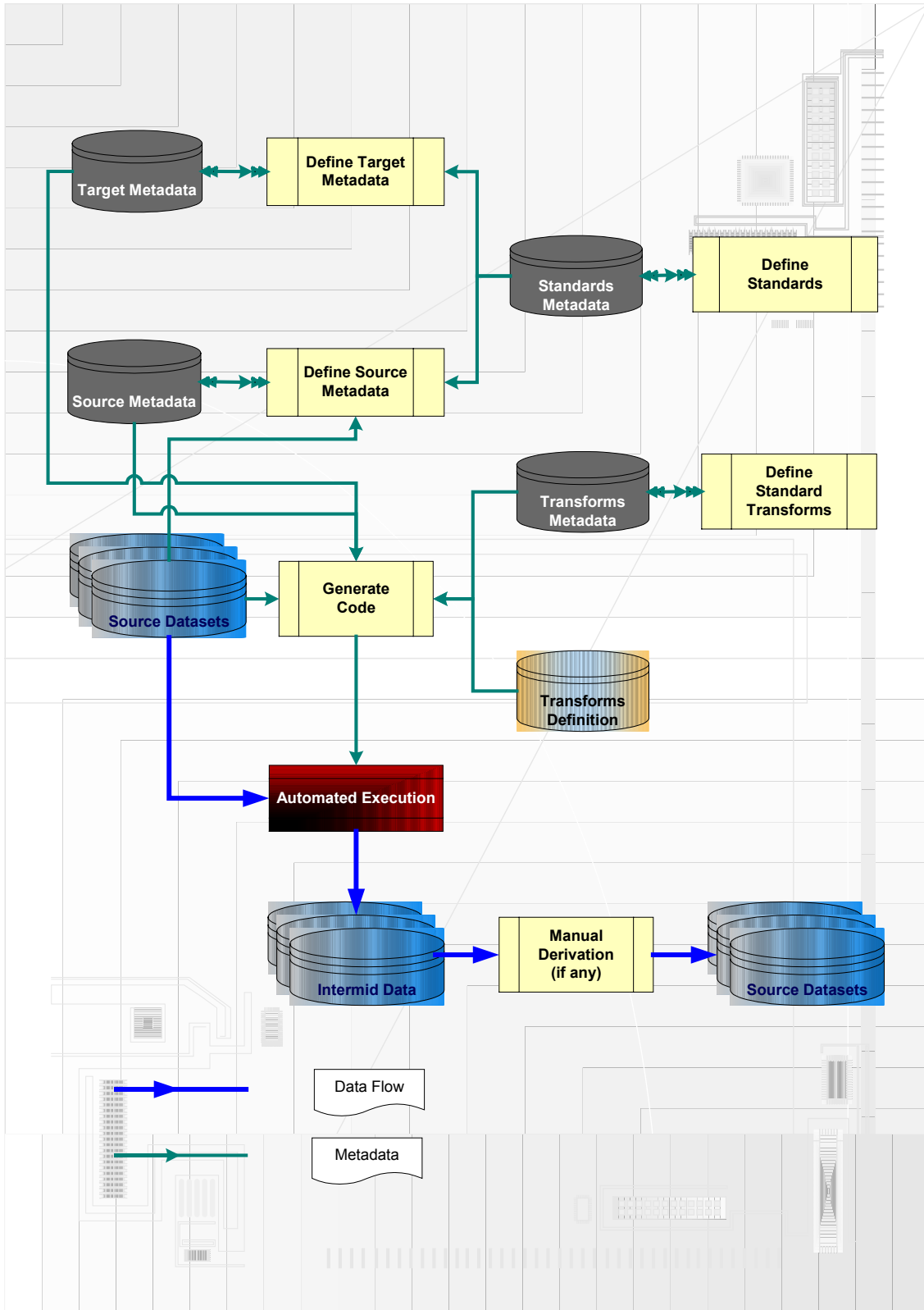


Figure 2: Framework

Description of the Process

The process creates and maintains metadata repositories and transformation definitions. The *Standards Metadata* and *Transforms Metadata* repositories are reusable across transformations. The *Source/Target Metadata* repositories are created for each transformation. The *Transforms Metadata* repository maps the source to the output for defined standards. These are implemented using the transformations defined in the *Transformation Definitions*. *Generate Code* process uses these inputs to generate SAS code that implements these transformations. Any additional processing required to customize the data according to customer needs, can be done using *Manual Derivation* process.

Explanation of Components

Source Metadata

Source Metadata would contain the metadata of the source database, pruned to the transformation needs. The 'Define Source Metadata' process would construct the metadata. It is to be defined for each transformation.

Define Source Metadata

The Define Source Metadata process would facilitate the creation of the source metadata. The source metadata can be defined by the taking complete metadata from the source library and pruning the un-required datasets and columns. Alternatively, if the source complies with a data standard, one can copy the metadata from the 'Standards Metadata' repository and then prune it. The benefit of using the 'Standards Metadata' is that the transformations can be picked from the 'Transforms Metadata' repository.

Standards Metadata

Metadata for some stable data standards can be stored in a repository. It can work as a re-usable component in defining target and source metadata. The database would probably have one row per variable per data standard.

Target Metadata

Target Metadata would contain the customized metadata of the target database. The 'Define Target Metadata' process would construct the metadata. It is to be defined for each transformation.

Define Target Metadata

The Define Target Metadata process would facilitate the creation of the target metadata. The target metadata can be defined by the taking proc contents of a similar target library and customizing the metadata to design the output database. Alternatively, if the intended target complies with a data standard, one can inherit its metadata from the 'Standards Metadata' repository and then customize it. The benefit of using the 'Standards Metadata' is that the transformations can be picked from the 'Transforms Metadata' repository.

Transforms Metadata

The transforms metadata would contain the mapping of variables from one data standard to another data standard. The possible columns of the metadata would be:

- Output Dataset
- Output Variable
- Output Standard
- Input Dataset
- Input Variable
- Input Standard
- Transformation Reference

The metadata would define a transformation to convert a source variable to a target variable.

Transforms Definitions

This would be the repository of the transformation definitions, which would be referred by the 'Transformation Reference' column of the 'Transforms Metadata'. These transformation definitions would be implemented as SAS macros.

Generate Code

The process would take input from the source metadata, target metadata, Transforms Metadata and Transforms Definitions to automatically generate SAS code, which would be applying all the applicable transformations to convert source database to the target database. The mapping of the input to the output and associating transformations shall be using the standards that define the source

and the targets. This process needs to be customized for the site of implementation.

Automated Execution

The SAS code generated in the 'Generate Code' process would be executed to produce the target database as defined in metadata for the transformation.

Manual Derivation (If Any)

In certain scenarios, it is not optimum to implement a transformation using the transformation infrastructure. In such a scenario, open code derivations can be applied to prepare the target database.

Benefits of the Framework:

The solution can be customized as per the need of the organization, allowing business users to state their requirements easily. The requirements themselves may be stated in set templates, thereby transforming itself into metadata by a validated process. Since a lot of code writing is being saved, there is actually a lot of saving on validation effort on open SAS code.

With the advent of powerful tools with SAS 9, this whole concept can be implemented easily, primarily through SAS ETL. However, there is a huge difference of cost in the two approaches. SAS ETL would require the setup of SAS ETL Server Technology solution. In our approach, the same can be achieved using front-end of SAS AF/SAS IntraNet with base SAS solution, even in version 8. The cost to build up the system is still much lesser than the cost involved in establishing and training cost of SAS ETL. We expect a cost difference of 50% on stable systems being used over a span of two years. Moreover, because of its design, of the application can be easily migrated to SAS9, in case the organization embraces SAS9.

Conclusion

The cost benefit and simplicity of use makes the architecture valuable to the SAS8 users interested in doing data integration. Though presented in the scenario of the clinical trial Pharmaceutical domain, the architecture can also support other SAS data integration projects.

Glossary / Abbreviations

SAS - Statistical Analysis System

CoE - Center of Excellence

ETL - Extract Transformation Load

CDISC - Clinical Data Interchange Standard Consortium

SAS InterNet - The technology, which enable web interface to SAS processing

SAS AF - The technology, which enable front-end windows with SAS

Acknowledgements

Rakesh Nandan, SAS CoE Lead in TCS, for providing valuable market intelligence.