

# Datamapper: A Documentation Generator for SAS Metadata

Lei Zhang, Merck & Co., Inc, Rahway, NJ

## ABSTRACT

SAS metadata is the data about SAS datasets, which is critical to the effective manipulation and analysis of SAS data. The SAS metadata exploration traditionally proceeds as ad-hoc programming on SAS Dictionary tables, which is inadequate, inefficient, and sometime complicated. Based on SAS ODBC functionality and hypertext techniques, a documentation generator called Datamapper is being developed to automatically create a set of structured hyperdocuments (called datamap) for the SAS metadata. The datamap allows users to interactively access essential piece of meta-information about SAS data objects or elements and traverse between them based on the established hyperlinks while avoiding ad-hoc SAS metadata programming. In this paper I first present a conceptual model for the SAS metadata and then the hyperdocument architecture of the datamap designed with OOHDM (Object Oriented Hypermedia Design Methodology), and finally discuss the development and implementation of the Datamapper.

## INTRODUCTION

SAS metadata is the data about SAS datasets. It is a very useful and indispensable resource for SAS program development and maintenance. A SAS program is a complicated software artifact that executes on multiple SAS datasets, and both developing it and maintaining it require that users have accurate, update-to-date meta-information about SAS datasets, such as SAS libraries, structures of datasets, variables, formats, and the relationships between them. For many large clinical trial projects, the meta-information involves in multiple data libraries with dozens of datasets and hundreds of SAS variables.

However, producing and keeping the SAS metadata up to date can be an expensive and time-consuming endeavor. Present approaches to SAS metadata are often inadequate to meet the diverse needs of both technical and non-technical users who often consult it on a day-to-day basis. In this paper, I introduce a tool called Datamapper that can automatically generate a collection of structured hyperdocuments called datamap for SAS metadata. Three unique features of the datamap are (1) that the datamap are created or updated mechanically and therefore never out of date. (2) that hyperlinks in the datamap are effectively created for a variety of data objects such as libraries, tables, formats, and variables. and (3) that hidden relationships and dependences under the SAS datasets are mined, recovered and reflected in the generated datamap. Therefore, surfing the datamap can often replaces ad hoc querying and programming about SAS metadata. Datamapper is a general-purpose utility tool for SAS metadata documentation, but many facets of its design are influenced by the goal of providing high quality and extremely effective metadata documentation for SAS datasets in clinical trial projects, which require metadata documentation be especially precise, accurate, and easy to use. This property is particularly important because it is essential that SAS programmers involved with the clinical trial projects be able to obtain *all* of the information that they need when they need it.

Presenting SAS metadata as hyperdocumentation raises a number of interesting problems in information retrieval and hypertext techniques. The evolving nature of SAS datasets and the diverse demands on metadata documentation presents a very challenging environment. This is made more challenging by the variety of data objects that must be integrated into a coherent set of hyperdocuments for the purpose of effective browsing or navigation.

This paper is organized as follows: Section 2 describes the problem domain and user requirements for SAS metadata documentation. Section 3 presents a conceptual model for the Datamap. The model is presented with UML class diagram. In Section 4, OOHDM is used to design the datamap navigational model. Both navigational space model and navigation structure model in UML are provided. In Section 5, HTML templates are used to present the model-based datamap. Section 6 describes the design and implementation of Datamapper. Finally, conclusion and plans for future work are presented in Section 7.

## THE PROBLEM DOMAIN

The typical working environment for most SAS programmers includes the SAS System, a couple of local or remote platforms, dozens of datasets stored in multiple directories, and little accessible metadata documentation about the data. The SAS System provides primitive support for the metadata of SAS datasets such as Dictionary tables, so a SAS programmer can write a piece of SAS codes or macros with Proc SQL and/or Proc Contents to capture the basic semantic and syntactical information about SAS datasets [1]. SAS Dictionary tables are helpful but have following limitations due to their machine-readable-only formats:

- The Dictionary tables are designed for efficient information retrieval and processing, not for efficient information representation. Without prior knowledge of their structures and organization, metadata information can not be accessed.
- Without proper SAS programming, the requested information can not be reached within a reasonable amount of time; The information retrieved through programming with Dictionary tables are ad-hoc, separated, and lack of context, and therefore hard to understand.
- The metadata contained the Dictionary tables are limited and insufficient to satisfy the variety of needs of SAS users; for example, Dictionary tables do not provide any information about missing values in a dataset. Finally
- The potential relationships and dependences between data objects are encrypted in the Dictionary tables. When the required metadata are retrieved, their associations with other data objects are lost, and uncovered. This situation often results in such assorted and scattered collection of metadata documents that many of them lie forgotten once used because the owner does not recall its existence, location, or content.

In this paper, a datamap concept is introduced to satisfy a variety of SAS metadata needs. The datamap is a collection of structural hyperdocuments for SAS metadata. It is designed to be an environment for users to browse SAS metadata and acquire varying levels of detail at the same time to get quick answers to specific questions. The datamap supports many potential users including the following:

- The SAS programmer who must develop SAS programs to manipulate datasets. This programmer will need detailed information on all related SAS libraries, tables, formats and variables and to determine the interactions between those objects.
- The SAS programmer who is new to the project assigned. This programmer will need high level and conceptual information presented in such a way that more detail is available when needed.
- The SAS programmer who maintains the SAS programs. This user needs detailed meta-information on given SAS

datasets to have a full understanding of the programs and datasets manipulated before making a change so that the SAS programs can be consistently maintained.

- The SAS program validator or reviewer who must validate the completed SAS programs. The validator or reviewer may need information at varying degrees of detail to complete his/her task.
- The statistician preparing a variety of statistical reports. This user needs relatively high-level yet very specific information on the areas of the reports covered.

This is a very rich spectrum of needs. To address these needs, the datamap is designed with OOHDM and is flexible enough to be automatically generated by Datamapper - a tool based on SAS ODBC, Java and HTML template techniques.

## A CONCEPTUAL MODEL FOR DATAMAP

The goal of datamap is to provide a "road map" for SAS datasets. It will help SAS users

- Locate and retrieve the data they are looking for, and understand the data they retrieved.
- Evaluate the suitability and appropriateness of the SAS datasets, in comparison with their initial needs.
- Merge different datasets from different sources, if needed, perform right operations on right datasets.
- Assess the data quality, and detect the anomalies or inaccuracies in the datasets; Pinpoint errors and omission before and after data manipulations.
- Facilitate more efficient communication between different users.

With the widespread use of the Internet, SAS users have become accustomed to the HTML interface as rendered by current browsers such as Internet Explorer and Netscape. Presenting SAS metadata in HTML format allows for platform-independence and easy access for both technical and non-technical SAS users with following advantages:

- It greatly simplifies the access to SAS metadata and minimizes the need of ad-hoc metadata querying and programming. SAS users who lack skill or time to explore the metadata directly can still apply information in their work.
- It makes available the huge amounts of SAS metadata that embedded in datasets in a way all SAS users can grasp instantly.
- It presents SAS metadata in a consistent and unified view of information: meta-information from different libraries or at different time can be combined into a single standard hyperdocumentation.
- It allows SAS metadata to be viewed locally or remotely when stored in a Web server, or be sent directly to many users through e-mail, resulting in effectively turning the metadata into multiple-user mode, at least for browsing.

The datamap aims to portray SAS metadata correctly and efficiently, and deemed particularly important to SAS programming and data analysis. It tries to collect and present five following categories of SAS metadata:

- **Semantics, Syntax and Structure**

Semantic SAS metadata is the information that is needed to describe SAS data such that it is interpretable by a user without sampling the raw data. At a less abstract term, semantic metadata is data description such as variable names and variable labels. Syntactical SAS metadata is used to describe the way that SAS data are stored and/or accessed. It usually consists of information about the data type of the SAS variables and access methods. Structural SAS metadata is used to describe the structure of groups of SAS variables, such as SAS libraries, tables etc.

- **Relational Metadata**

Semantics, syntax and structure are entities or attributes used to describe one or more data objects in SAS datasets. Another very important issue is the associative relationship between those data objects. Association SAS metadata denote such information that is used to locate data objects and sources of interest.

- **Similarity, Dissimilarity and Naming Equivalence**

Since the structure of SAS datasets will change or shift over time, and the data may be produced from different sources by different projects or organizations, the information about the similarity, dissimilarity and naming equivalence between SAS libraries are very useful for reusing both programming codes and data analytic methods.

- **Statistical Information about SAS Variables**

Statistical information about SAS variables is helpful for massaging data for more effective analysis and data quality check: For example, the discovery of outliers and missing data points. Statistical information about a SAS variable includes distinct values that the variable has, frequency of the distinct values, range, extreme values, and the number of missing values so on. It is not only very useful for data analysis and assessment of the data quality, but also for coding, testing and debugging of SAS programs.

- **Illustration, Explanation and Clarification**

Apart from the definitions of data objects, presenting samples about the data objects as illustrations and explanations is very useful to reinforce the need for several layers of on-demand explanation and elaboration for tabled data. For example, a sample table from a dataset will give a user a more clear understanding of structures and relationships between SAS variables within a dataset.

Based on the above requirement, a UML representation of datamap for the SAS metadata is given in Fig 1.

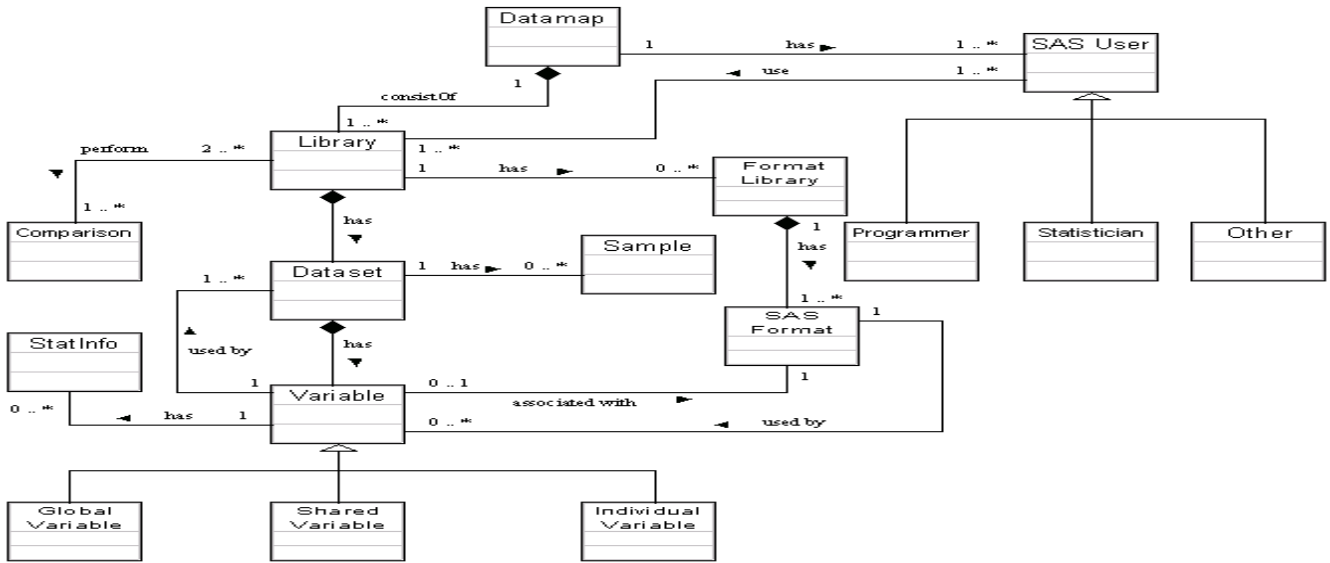


Figure 1. A UML Class Diagram for SAS Datamap

### NAVIGATIONAL DESIGN FOR DATAMAP

The navigational design is a critical step in the development of hypertext applications like datamap. According to OOHDM [2], the navigational design is represented with two UML models: the *navigational class model* and *navigational structure model*. The navigational class model defines a view on the conceptual model that shows which classes in the conceptual model can be viewed through navigation in a hypertext application. A navigational class is defined as a stereotyped class <<navigational class>> with the same name as the corresponding class in the conceptual model. Navigational objects are instances of these navigational classes connected by links (in UML terms) that are instances of the associations of the navigational model. The navigational class model is usually a navigation-oriented conceptual model where

the classes and associations, which are not needed for navigation, are eliminated or reduced to attributes of other classes.

The navigational class model for datamap is shown in Fig 2. All classes and associations in the conceptual model are included in the navigational model with the exception of SAS User class and associations that link SAS user to datamap because they are irrelevant in the hypertext navigational process. Additional associations between Datamap and Variable classes are added for direct navigation in order to avoid length of main navigation paths starting from Datamap class greater than one.

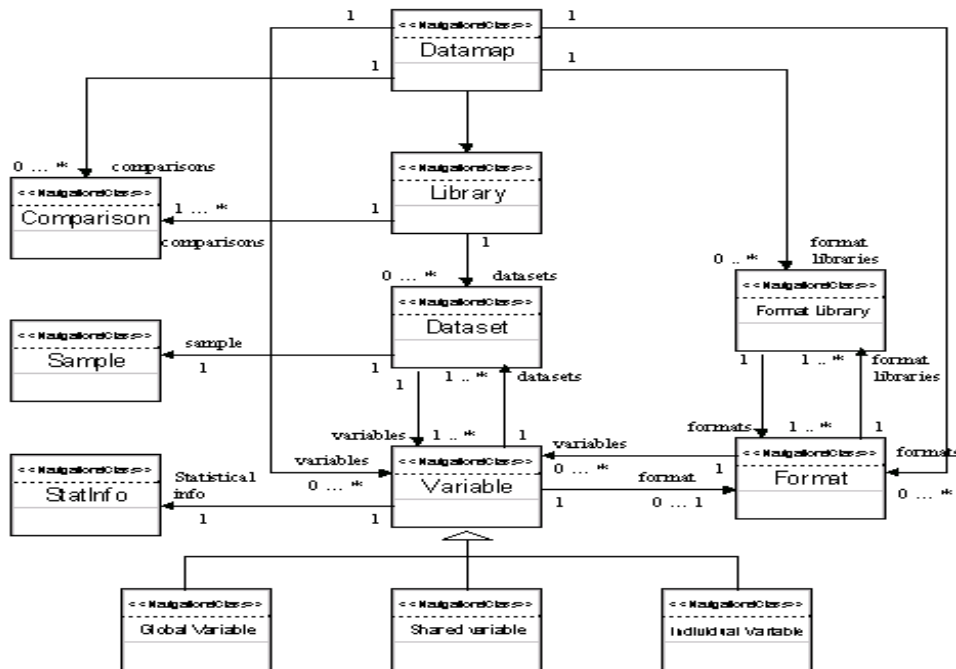


Figure 2. Navigation Class Model for SAS Datamap

With the navigation class model above, a navigation structure model is created to describe how the navigation can be performed using access element like indexes, guided tours,

queries and menus. The navigation structure model is showed in Fig 3, in which two types of access elements, index and menu are widely used.

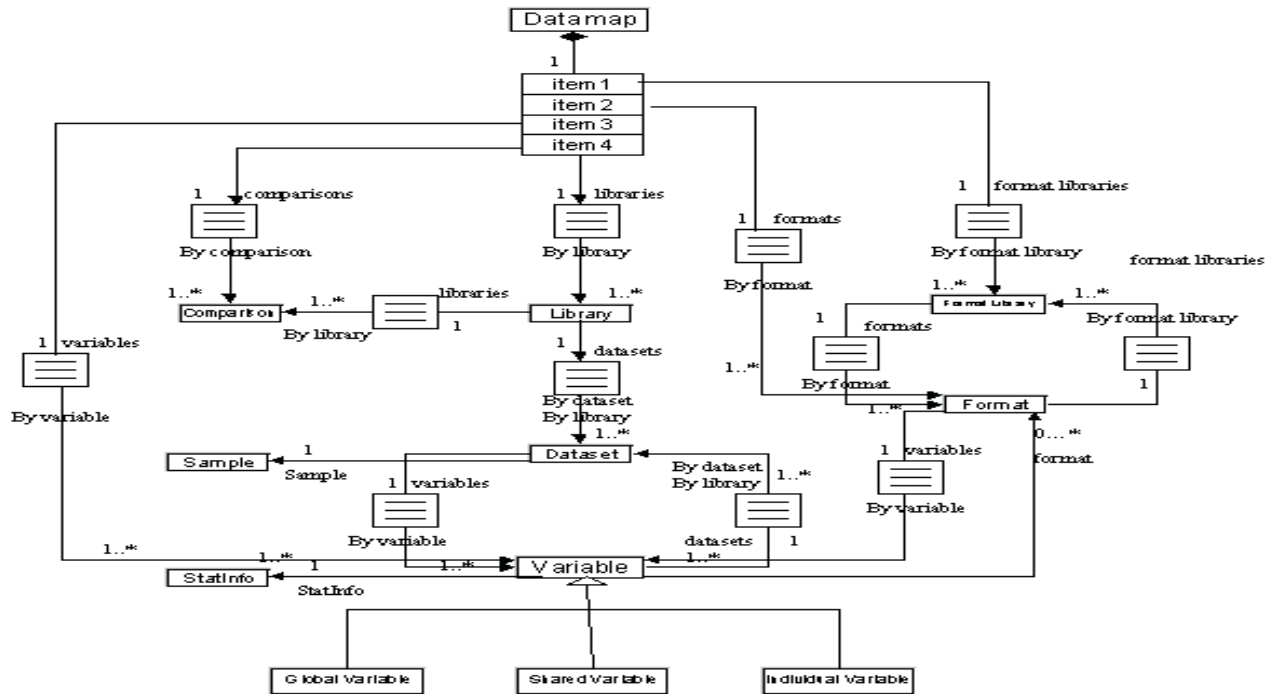


Figure 3. Navigation Structure Model for SAS Datamap

### PRESENTATION DESIGN FOR DATAMAP

With the navigational model in the previous section, it is decided that the presentation of datamap is divided into two parts: one part provides a presentation of navigation context, which shows users the actual navigation paths and entry points, and the other

part shows the corresponding contents. HTML framesets are used to allow users to visualize both navigation structure and contents separately, because framesets provide an ideal way to preserve and present the relationships between the navigational objects. The Datamap navigation class is selected as the root of navigation. Figure 4 is a sample datamap generated by Datamapper.

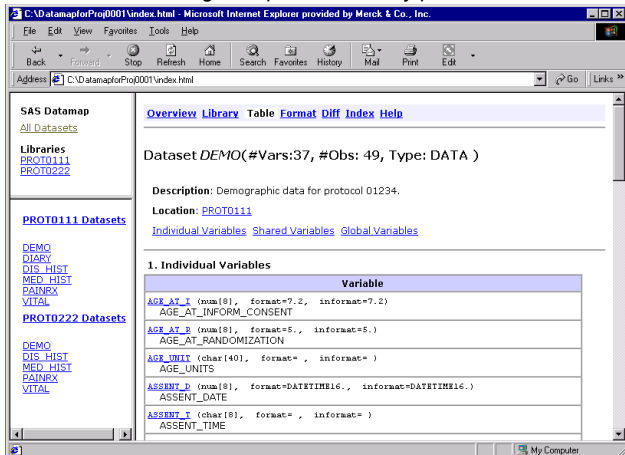


Figure 4. Datamap Screenshot

As can be seen in Figure 4, the screen is consisted of three frames. The left two frames consist of the table of contents, which is the navigation device for the datamap. From the table of contents on the left, users find their way among the datasets using hyperlinks. The underlying library structure of a SAS data source is the basic indexing model for the datamap. The libraries and datasets are listed alphabetically; libraries in the upper left frame and datasets in the lower frame. The dataset list also contains information about the library the dataset belongs to.

Clicking on an active library name brings a new dataset list to the left lower frame. Clicking on an active dataset name loads the dataset document into the right-hand main frame. There is also a hyperlink that point to a document containing all datasets ordered alphabetically in the upper left frame. The right-hand main frame contains the contents. The table or dataset documents are its core contents. The purpose of the dataset document is to describe the dataset attributes and its relations to other objects such as libraries, variables and formats. Hypertext is used to link from the dataset documents either to the related documents or to other parts of the document. Dataset documents have a header and a footer that contain hyperlinks or menus to other documents, and to strategic points in the dataset documents

(See Figure 5). Apart from hyperlinks to SAS Libraries and Formats, users can also click on Diff menu to compare the difference between two SAS libraries, and click Index menu to search a whole list of SAS variables in alphabetical order. The same header and footer are available in all documents displayed in the main frame but links are not always active.

[Overview](#) [Library](#) [Table](#) [Format](#) [Diff](#) [Index](#) [Help](#)

Figure 5. The Menu Items in the Right-hand Main Frame of SAS Datamap

After the header comes a general description of the dataset and the hyperlink to a sample data table for the dataset, and then

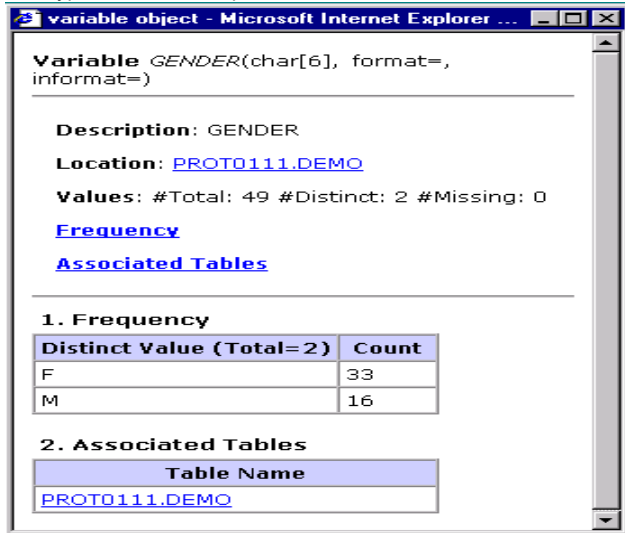


Figure 6. Detailed Document for A Variable in a Dataset

The HTML files for a datamap are stored in several different directories within a common root directory. This directory structure is designed to reflect the way in which datamap hypertext files are typically organized for navigation and indexing. All directories can have nested sub directories and files.

The content structure and typography of the datamap is defined by a collection of HTML templates. A HTML template is a formatted collection of HTML document skeletons that can be filled in by a program or a user. It is a technique that adds parameter capabilities to static HTML documents by embedding either scripting codes or custom tags. The major advantage of using HTML template is separating HTML presentation of navigational classes from data and codes that generate the HTML pages. A HTML template normally corresponds to a navigational class in the datamap navigational model. There are several templating techniques available. I use HTML.Template for Java, a freeware developed by Philip S Tellis [4], as a template engine to support the presentation and implementation of Datamap.

HTML.Template for Java extends HTML with a very small set of HTML-like tags - `<tmpl_var>`, `<tmpl_if>`, `<tmpl_unless>`, `<tmpl_loop>` and `<tmpl_include>`, which provide variable substitution, looping and branching. A HTML template for a navigational class is consisted of HTML and these new tags. Below is an example of HTML template for library Index access element in the datamap navigational structure model.

comes the summaries of three classes of variables within this dataset: Global SAS variables, which appears in all datasets in a library; Sharable SAS variables, which appears in more than one datasets in a library; Individual SAS variables, which only appears in this datasets. Each summary contains primary information about categorized variables, such as variable name, type, and format currently associated. If a more detailed information is needed, a user can click on the active variable name to open up a small window which shows the information about the variable such as distribution of the values, missing values it has, associated tables that contain the same variable name, and so on (See Figure 6). The dataset document ends with footer that contains the same hyperlinks as its header.

```
<html>
<head>
<title> Library Index </title>
</head>

<body>
<table>
<Caption align=top><B> SAS
Libraries</B></Caption>
<TD>
<TMPL_LOOP LibIndex><Br>
  <a href="Index/<TMPL_VAR LIBNAME>.html"
    target="tableframe"><TMPL_VAR LIBNAME></a>
</TMPL_LOOP>
</TD>
</table>
</body>
</html>
```

In the next section, I will describe how Datamapper use the HTML.templates to generate a datamap.

### A TOOL FOR AUTOMATED GENERATION OF DATAMAP

Datamapper is an authoring tool, written in Java, which implements the datamap model described in previous sections. It automatically generates a datamap for a given SAS ODBC data source, and has very simple GUI interface implemented with Java/Swing (See Figure 7).

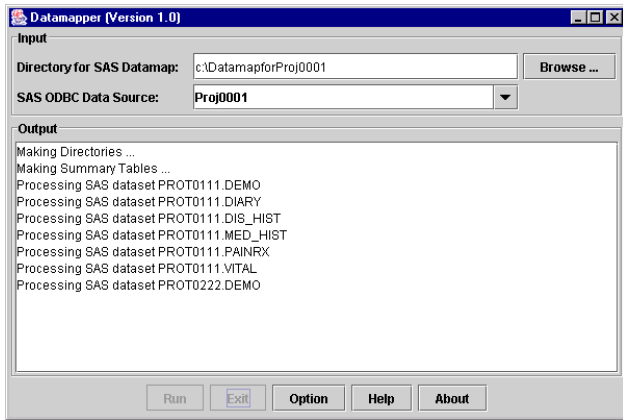


Figure 7 The Screenshot of Datamapper

Datamapper takes two inputs: the first is the SAS ODBC data source name (DSN), and the second is the name of the directory that stores the datamap to be generated. Since the SAS ODBC data source can register multiple SAS libraries, Datamapper can be used to create a datamap for any combination of SAS libraries.

The purpose of Datamapper is to provide a convenient tool with which SAS programmers and statisticians can create standard datamaps with a few clicks. The automated generation of SAS

Datamap is critical, because (1) many SAS programmers and statisticians, especially those involved with clinical trial data analysis, often don't have additional overhead to service SAS metadata requirements, not to mention of composing hypertexts for SAS metadata; (2) There is little or no margin to increase staff costs to fund the compiling of the SAS metadata documents.

Datamapper is actually not one program but a complex structure of systems, which is illustrated in Figure 8.

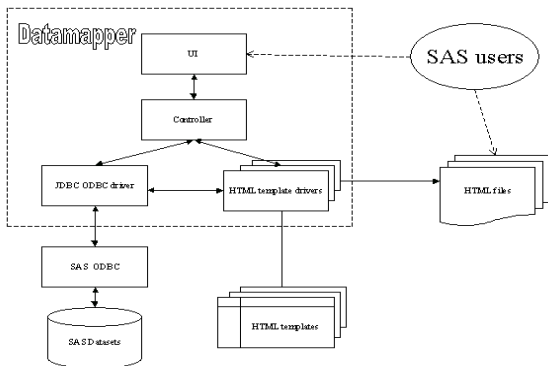


Figure 8 The Architecture of Datamapper

As seen from Figure 8, Datamapper is consisted of Controller, HTML template drivers, JDBC-ODBC driver, and UI module. The role of Controller is to build an information structure and hand over the process of generating output to HTML template drivers. A HTML template driver is a Java class that dynamically binds a HTML template with specified metadata that are retrieved from SAS ODBC data source through Java JDBC-ODBC driver and generate the corresponding HTML document (for accessing to SAS ODBC functionality, please see [5] for more details). For example, Figure 9 is a driver class for the HTML template of the library Index access element described in the previous section.

```
import java.io.*;
import java.util.*;
import java.sql.*;
import HTML.Template;
public class LibIndexTemplateDriver
    extends TemplateDriver {

public LibIndexTemplateDriver(Connection con,
    String templateFile, String htmlFile) {
    super(con, templateFile, htmlFile);
}

public void makeHTML() {
```

```
try {
// Create the HTML Template object
Template template
    = new Template(templateFile);

//Get a list of different library names
//excluding system default library MAPS,
//SASHELP, SASUSER and SASADMIN.
SQLExecutor sel = new SQLExecutor(con,
    "select distinct libname from " +
    "dictionary.tables where libname not in " +
    " ('MAPS', 'SASHELP', 'SASUSER',
        'SASADMIN') order by 1");
ResultSet rs = sel.getResultSet();

// Set parameters
Vector libIndex = new Vector();
Hashtable libItem;

while(rs.next()) {
    libItem = new Hashtable();
    libItem.put("LibName", rs.getString(1));
    libIndex.addElement(libItem);
}

template.setParam("LibIndex", libIndex);

// Generate HTML files.
save(template.output(), htmlFile);
```

```
    } catch(Exception e) {  
    }  
}  
}
```

Using HTML template technique greatly simplifies the automated generation of datamap because of the separation of metadata representation and typography from the metadata contents.

### **CONCLUSIONS AND FUTURE WORK**

This paper describes a hypertext model of the extended SAS metadata and the Datamapper tool that automatically generates the metadata documentation for the given SAS data source based on the datamap model and SAS ODBC functionality. The tool provides a convenient mechanism for SAS users to obtain standardized SAS metadata documents and browse them locally or remotely the way they surf Web without requiring knowledge of mundane details about SAS datasets and additional SAS programming.

Future goals for the Datamapper tool is the development of the ability to provide more user-friendly, consistent navigational features, the ability to create multiple views on a given SAS data source for different type of SAS users, and the ability to integrate datamap with SAS source code hypertext documents.

### **REFERENCES**

1. Pete Lund, "A Quick and Easy Data Dictionary Macro", Proceedings of SAS Users Group International Conference 2002 (SUGI 27), Orlando, Florida
2. The Object-Oriented Hypermedia Design Model (OOHDM) Home Page. <http://www.telemidia.pucrio.br/oohdm/oohdm.html>
3. Priestley, M. "Navigation Issues in Hypertext: Documenting Complex Hierarchies with HTML Frames", Proceedings of the 15th Annual International Conference on Computer Documentation, 1997, 223-235
4. HTML Template for Java Home Page. <http://html-templ-java.sourceforge.net>
5. Lei Zhang and Tianshu Li, "Using SAS ODBC with Java", Proceedings of Northeast SAS Users Group 15<sup>th</sup> Annual Conference 2002 (NESUG 15), Buffalo, New York

### **ACKNOWLEDGMENTS**

The following people contributed extensively to the development of this paper: Izabella Peszek, Elaine Czarnecki, and John Troxell at Merck & Co. Inc.. Their support and encouragement is greatly appreciated.

### **CONTACT INFORMATION**

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

Lei Zhang  
Merck & Co. Inc.  
RY34-A320  
P.O. Box 2000  
Rahway, NJ 07065  
(732) 594-9856  
(732) 594-6075 (Fax)  
[Lei\\_zhang4@merck.com](mailto:Lei_zhang4@merck.com)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® Indicates USA registration.

Other brand and product names are trademarks of their respective companies.

