Paper 031-29

A Lightweight HTML Codebook Generator for Clinical Trial Data

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

ABSTRACT

A codebook is a set of technical documents about the datasets collected for a particular purpose. It is the metadata, or information about the data. Clinical trial data are often complicated and highly interrelated, a well-organized dynamic codebook for them, therefore, is of vital importance to both clinical SAS programmers and statisticians who need explore and utilize clinical datasets for comprehension and communication. In this paper I first introduce the concept and structure of the HTML codebook model specially designed for clinical datasets, and then present a software generator called Datamapper that can automatically create HTML codebooks from SAS datasets with the HTML templates derived from the conceptual model. The Datamapper, implemented with SAS ODBC, HTML templates and Java technologies, requires minimum user inputs and configuration in the SAS System. The HTML codebook (also called datamap) it generated allows clinical SAS programmers, statisticians, or any other users to interactively access essential piece of information about clinical data objects and navigate over them like a web surfer.

INTRODUCTION

The codebook for clinical datasets is frequently needed at different phases of clinical data analysis in order to grasp the clinical trial data at a certain level of abstraction within a short amount of time. For instance, a codebook can be used to quickly find and interpret variables in a dataset while programmers are writing their SAS codes. It is also valuable in mapping functional query requests as expressed by clinical statisticians onto technical query requests, and in estimating the time and cost of such clinical query tasks. In addition, a codebook will greatly facilitate the validation and maintenance process of clinical SAS programs. A good codebook usually supports many potential users in a clinical trial project including the following:

- The SAS programmer who is responsible for writing SAS programs to analyze the clinical trial data. This programmer usually wants to acquire deep knowledge on the data elements of interest, such as data libraries, datasets, variables, and formats, and the relationships and interactions among those data elements.
- The SAS programmer who is new to the clinical trial project just assigned. This programmer usually need high level and conceptual information at the beginning and wants to obtain more at later stages.
- The SAS programmer who maintains existing SAS programs for a couple of clinical projects. This user often needs a quick way to refresh his/her mind on the programs and datasets that haven’t been worked on for quite a while when he/she is requested to modify them due to bug fixing, adding new functions, and so on.
- The SAS program reviewer who must verify and validate the SAS programs written by others. The reviewer needs independent information sources about clinical datasets that SAS programs used at varying levels of detail while he/she walks thought, tests, or even writes his/her own programs to compare with the SAS programs being validated.
- The statistician who is responsible for planning and monitoring SAS programs for a variety of clinical analysis reports. This user needs relatively high-level yet very specific information on the different aspects of clinical datasets that individual report covered.

WHY IS THERE A NEED FOR AN HTML CODEBOOK FOR CLINICAL TRIAL DATA?

The typical working environment for most clinical SAS programmers and statisticians includes the SAS System, a local PC with connections to multiple larger platforms, dozens of clinical datasets with hundreds of variables stored in several SAS data libraries. If they are lucky enough, a MS Word or Excel file will be provided to serve as a static codebook that roughly describes the column structure and meanings of the clinical datasets to be worked with. In most situations, SAS programmers have to write their own ad-hoc SAS codes with Proc Contents, Proc Datasets, or Proc SQL statements on SAS system dictionary tables to dig up the meta-information about the datasets of interest. As we all know, clinical trial data are often highly interrelated. Their structure and contents may change over time and usually contain lots of missing values, abnormal data points and unexpected outcomes. Clinical SAS programmers or statisticians usually do not have enough time to comprehend all the facts buried in the large amount of clinical trial data. They tend not to read the whole codebook, if there is one, from beginning to end but frequently jump between related data elements of interest. Therefore, they prefer to have a dynamic codebook to provide enough information so that they can build a mental model of the data, and zoom in to the specific details they are interested in at certain time. The level of details they are interested in depends very much on what they intend to do with the clinical datasets. This kind of flexibility requirement is very hard to be achieved with common MS Word or Excel files.

With the widespread use of the Internet, people have become accustomed to the HTML interface as rendered by popular browsers such as Internet Explorer or Netscape. Presenting a codebook in HTML format allows for platformindependence and easy access for both technical and non-technical users of clinical trial data. If an HTML codebook can be systematically generated from a given SAS data source with little or no effort, it will have the following extra advantages:

- It will greatly simplify the access to meta-information about clinical SAS datasets and minimizes the need of ad-hoc metadata querying and programming. Users who lack skill or time to explore clinical datasets either directly or indirectly can still apply the information in their work.
- It will provide different levels of abstraction about the clinical trial data in a standard way. Users can move smoothly from one level of abstraction to another, without losing their position in the codebook (zooming in or zooming out), which makes available the huge amounts of meta-information that are embedded in clinical datasets in a way all users can grasp easily and instantly.
- It will present a consistent and unified view on meta-information about clinical trial data across different SAS data libraries, and even across clinical projects at different time. Users will have a convenient and intuitive way to look into the difference and similarity between variables, datasets, data libraries and even clinical projects.
- It will make effective information sharing and communication among different team members at different time and location. Standardized HTML codebooks can be viewed locally or remotely when stored in a Web server, or be packed and sent directly to many users through e-mail with little or no extra documenting effort.

THE HTML CODEBOOK MODEL FOR CLINICAL TRIAL DATA

Clinical trial data are typically presented in chronological order, organized around clinical trial concepts such as patient demographics, medical history, vital signs, lab tests, adverse events, and so on. An HTML codebook should be designed as a digital "road map" for the clinical datasets. It should not only describe key SAS metadata elements or objects such as variable, dataset, and data library which can be obtained from SAS system dictionary tables, but also expose the potential domain-oriented relationships between those key objects and offer different views on data behind them. It should assist users in data cleaning, data transformation, data reduction, and data mining and knowledge representation in the whole life cycle of clinical trail studies. Based on the characteristics of clinical datasets and practical requirements, a Unified Modeling Language (UML) representation of main conceptual objects in the HTML codebook is given in Fig . 1.

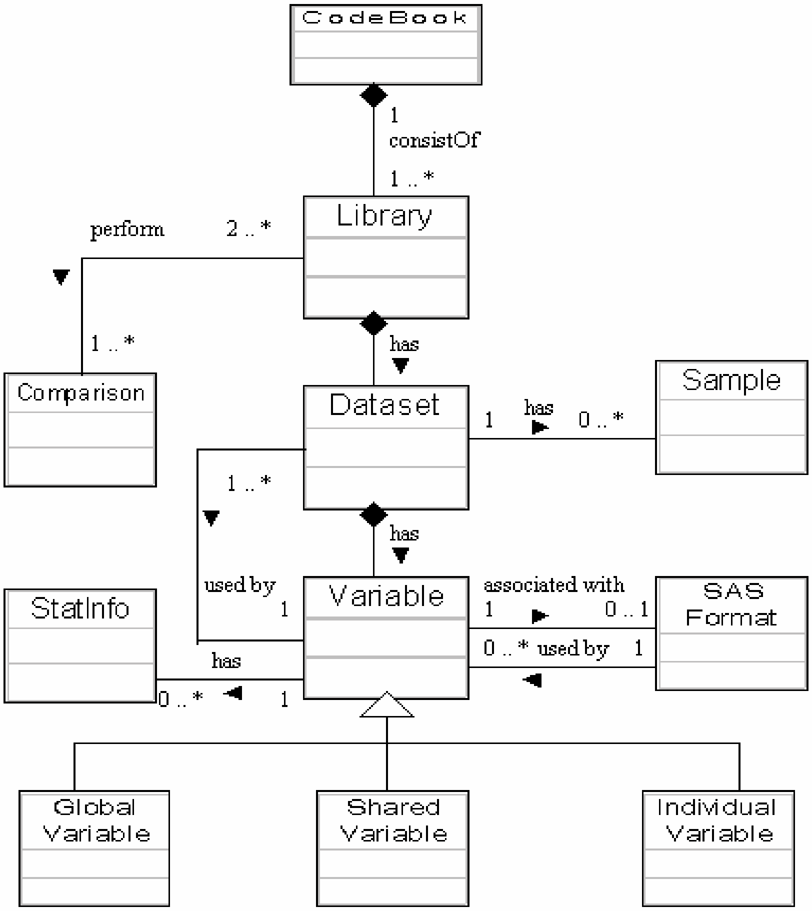


Fig. 1 An HTML Codebook Model for Clinical Trial Data

Due to the limitation of the space, here I only summarize the meta-information for three key objects that is variable, dataset and library in the above UML model.

Variable object in the UML model provides following information about a variable in clinical datasets.

- Syntax and structure: variable name, label or description, type, length, and location in the data library.
- Associations with other objects: including associated format and informat, links to the datasets that contain the same variable name.
- Data properties of this variable: including statistical information about the data points of the variable such as number of distinct values, number of missing values, frequencies for the distinct values, etc.

The information about the syntax,

structure and association of a variable can be obtained from SAS dictionary tables while its data properties can be extracted from the data points of the variable.

Dataset object in the UML model provides following information about a dataset in a clinical data library.

- Syntax and structure: dataset name, label or description, dataset type, variables for patient ID and visit ID if available, etc. A clinical dataset can be classified into following five types based on patient ID and visit ID variables and observations in the dataset:
  1. A type 1 clinical trial dataset has the data collected once for each patient. The data are not particularly related to patient visits. For example, demographic data for each patient.
  2. A type 2 clinical trial dataset has the data recorded multiple times for each patient. The data are not particularly related to patient visits. For example, previous medication history for each patient.
  3. A type 3 clinical trial dataset has the data collected once for each patient and visit, For example, vital signs collected for each patient at every visit.
  4. A type 4 clinical trial dataset has the data collected multiple times for each patient and visit. For example, lab tests collected for each patient at every visit.
  5. A type 5 clinical trial dataset has the data that is not related to any particular patient or visit. For example, a list of laboratory normal ranges, coding dictionaries, system tables, or derived summary datasets.
- Classification of variables in the dataset: all variables in a dataset are organized into three categories:
  - Individual variables with their names occurring only in this dataset,
  - Shared variables with their names occurring in multiple but not all datasets in a data library, and
  - Global variables with their names occurring in all datasets in a data library.
- Associations with other objects: including links to the associated variables, and associated library.
- Visit flow chart that reflects the changes in number of patients in each visit within this dataset.
- Statistics on this dataset: including number of variables, and number of observations.
- Sets of sample observations that show what typical data observations look like in the dataset.

The basic meta-information about a dataset can be obtained from SAS dictionary tables. The rest of the meta-information such as dataset type, visit flow chart can be derived from dataset with the patient ID and visit ID variables user provides.

Library object in the UML provides the following information about a data library in clinical trial data.

- Library name
- Total patient population, and the total number of patients in each visit.
- Summary of datasets within this library, including dataset type or pattern, number of variables and observations in each datasets, etc.
- Visit distribution over datasets within this library.
- Comparison with other libraries

Most of meta-information listed above can be derived from SAS dictionary tables and datasets under the data library.

In the navigation design of the HTML codebook discussed in next subsection, the objects defined in the above conceptual model will be used to derive nodes or HTML pages of HTML codebooks while the associations will be used to derive the links between HTML pages.

NAVIGATION AND PRESENTATION OF THE HTML CODEBOOK

Since the HTML codebook is aimed to help users to construct a mental model that represents the clinical data objects and their semantic relations, the heart of the HTML codebook lies in its navigation using hyperlinks and HTML pages, The navigational design should reveal when, where, and how data objects in the previous conceptual model can be viewed through navigation in the HTML codebook. The Object-Oriented Hypermedia Design Model (OOHDM) methodology described in [2] is used to present the conceptual navigational paths within HTML codebooks. Fig. 2 shows the navigation-oriented UML model for HTML codebooks.

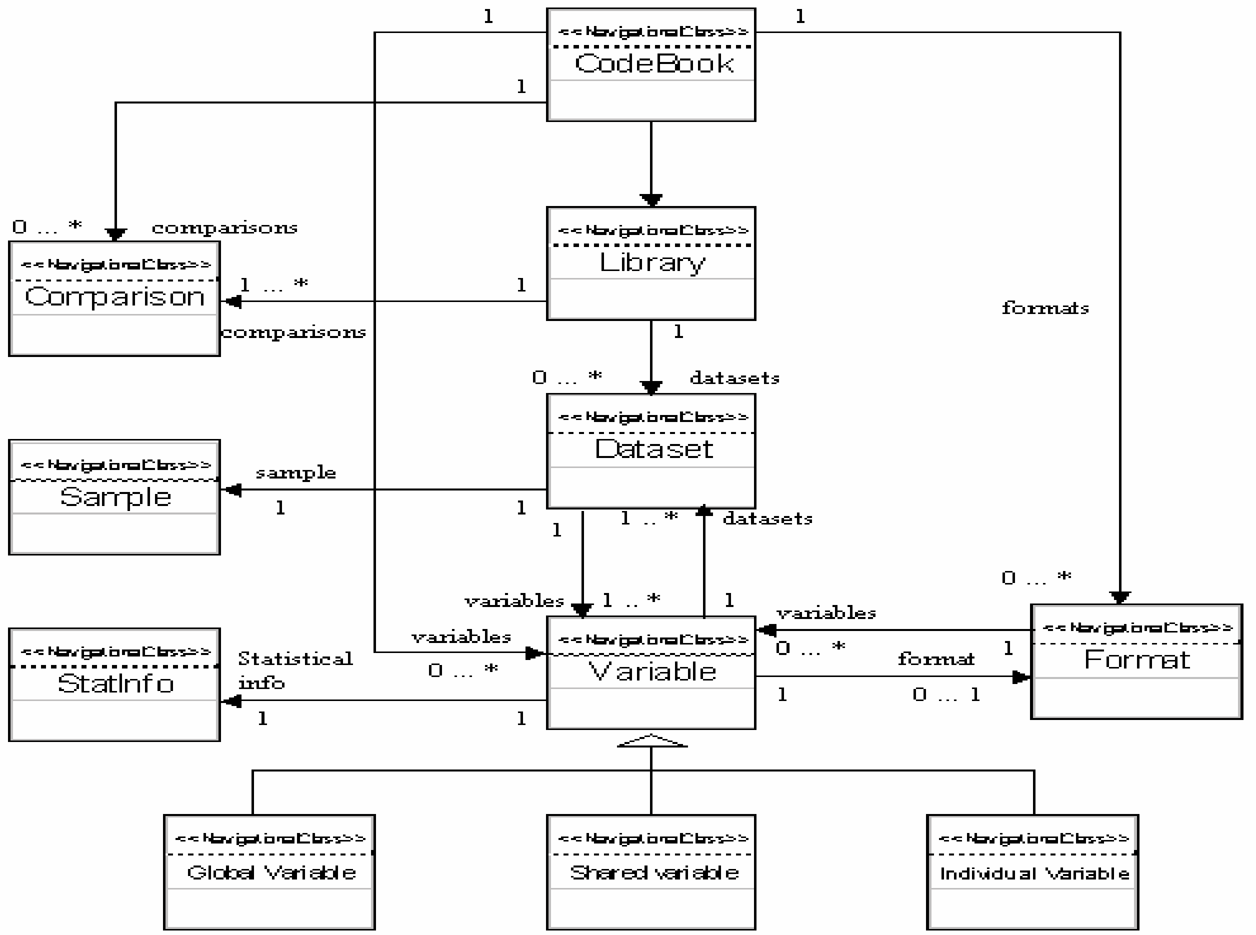


Fig. 2 A navigation oriented UML model for the HTML codebook

The navigation-oriented model defines a view on the UML conceptual model in Fig. 1. It includes all objects and associations in the previous UML conceptual model. Additional paths between codebook object, variable objects, and format objects are added for direct navigation in order to avoid lengthy key navigation paths starting from codebook object.

With the navigation-oriented model, a navigation structure model for the HTML codebook is created to describe how the navigation can be performed using various access elements like indexes, guided tours, and menus. The navigation structure model is showed in Fig. 3, in which two types of access elements, index, inverted index and menu are widely used.

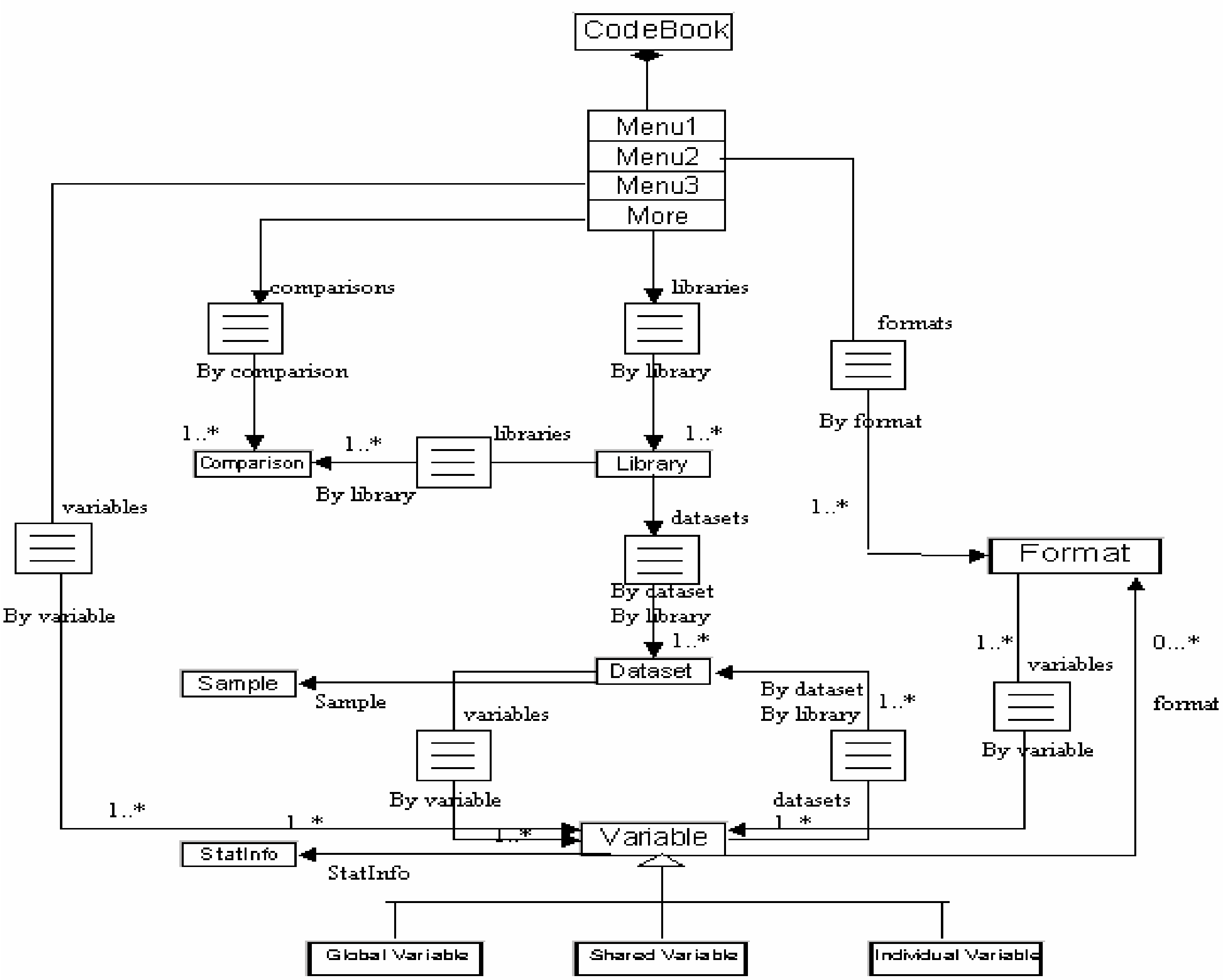
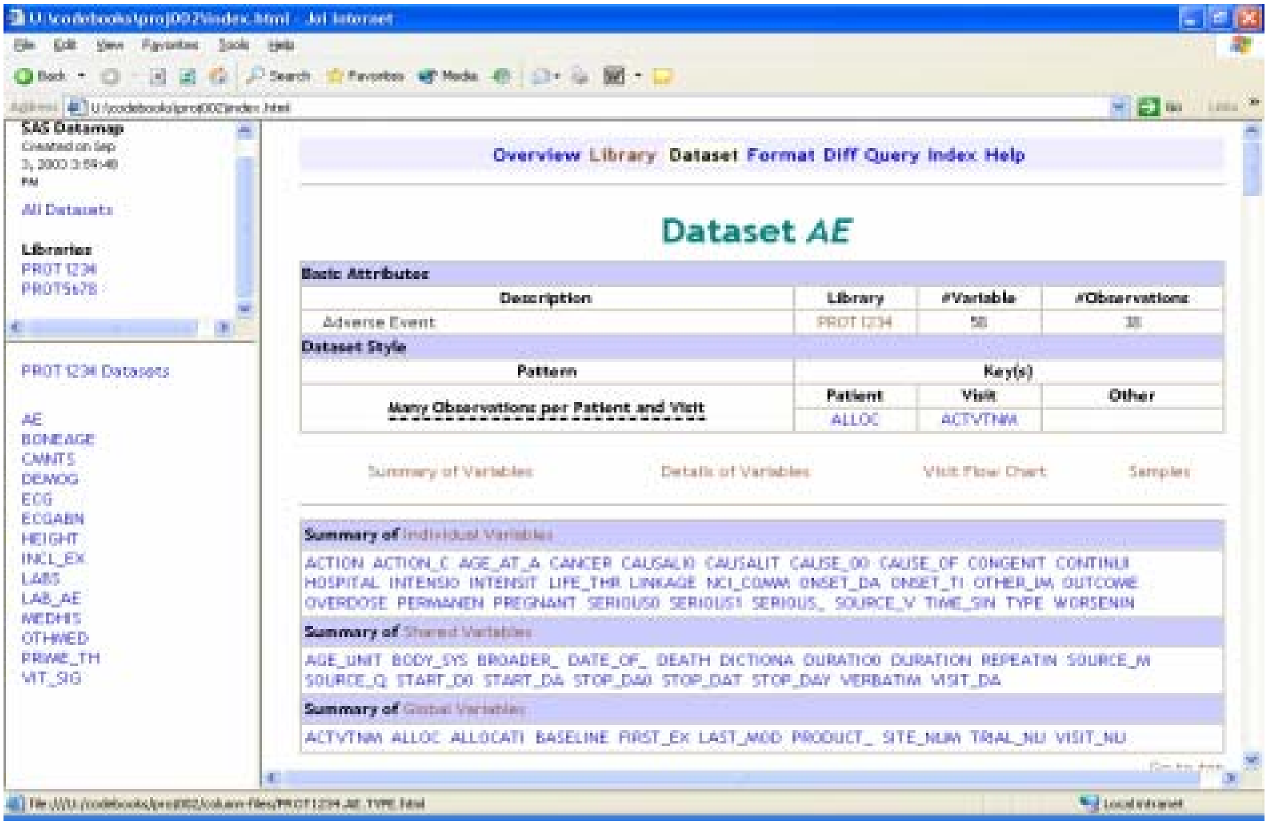


Fig. 3 A navigation structure model for the HTML codebook

With the navigational model described above, the presentation of the HTML codebook is divided into two parts: one part presents navigation context, which shows users the actual navigation paths and strategic entry points, and the other part displays the corresponding contents. Since HTML framesets provide an ideal way to preserve and present the relationships between the structured navigable objects, they are used in HTML codebooks to visualize both navigation structure and contents separately and simultaneously. Fig. 4 is the screenshot of a sample HTML codebook for a set of clinical datasets.


U:\codebooks\proj002\index.html - Jol Internet
File Edit View Favorites Tools Help
Back Search Favorites Media
Address U:\codebooks\proj002\index.html
Go Links
File:///U:/codebooks/proj002/column-files/PROT1234_AE_TYPE.html
Local intranet

SAS Datamap
Created on Sep 3, 2003 3:59:40 PM

All Datasets

Libraries
PROT1234
PROT5678

PROT1234 Datasets

AE
BONEAGE
CMNTS
DEMOG
ECG
ECGABN
HEIGHT
INCL_EX
LABS
LAB_AE
MEDHIS
OTHMED
PRIME_TH
VIT_SIG

Overview Library Dataset Format Diff Query Index Help

Dataset *AE*

Basic Attributes			
Description	Library	#Variable	#Observations
Adverse Event	PROT1234	58	38

Dataset Style			
Pattern	Key(s)		
	Patient	Visit	Other
Many Observations per Patient and Visit	ALLOC	ACTVTNM	

Summary of Variables   Details of Variables   Visit Flow Chart   Samples

Summary of Individual Variables

ACTION ACTION_C AGE_AT_A CANCER CAUSALIO CAUSALIT CAUSE_00 CAUSE_OF CONGENIT CONTINUI HOSPITAL INTENSIO INTENSIT LIFE_THR LINKAGE NCI_COMM ONSET_DA ONSET_TI OTHER_IM OUTCOME OVERDOSE PERMANEN PREGNANT SERIOUS0 SERIOUS1 SERIOUS_ SOURCE_V TIME_SIN TYPE WORSENIN

Summary of Shared Variables

AGE_UNIT BODY_SYS BROADER_ DATE_OF_ DEATH DICTIONA DURATIO0 DURATION REPEATIN SOURCE_M SOURCE_Q START_D0 START_DA STOP_DA0 STOP_DAT STOP_DAY VERBATIM VISIT_DA

Summary of Global Variables

ACTVTNM ALLOC ALLOCATI BASELINE FIRST_EX LAST_MOD PRODUCT_ SITE_NUM TRIAL_NU VISIT_NU

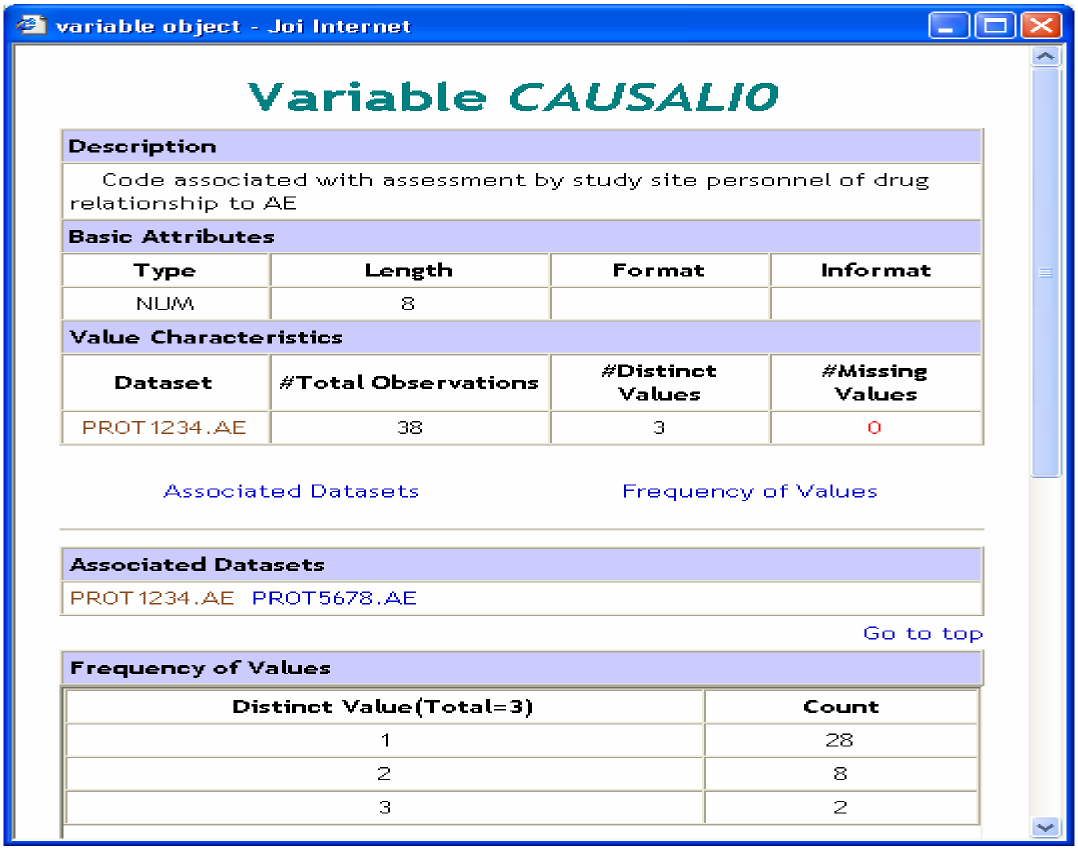
Go to top

Fig. 4 The screenshot of a sample HTML codebook

As can be seen from Fig. 4, the browser window is consisted of three frames. The left two frames consist of the table of contents, which is the main navigation device for the HTML codebook. From the table of contents on the left, users can find their way among the datasets with the hyperlinks in the left two frames. The libraries and datasets are listed alphabetically; libraries in the upper left frame and datasets in the lower frame. The dataset list contains information about the library the dataset it belongs to. Clicking on a library name brings a new list of datasets to the left lower frame. Clicking on a dataset name loads the dataset HTML page into the right-hand main frame. The right-hand main frame mainly shows the library or dataset HTML pages under the current context. They describe the dataset/library attributes and their relations to other objects such as variables and formats. Embedded hyperlinks link from the current dataset/library pages either to the related HTML pages or to other parts of the HTML page. All HTML pages showed in the main content frame have a header that contains hyperlinks or menus to other HTML documents, and/or to strategic entry points in HTML codebooks (See Fig. 5 below). Apart from hyperlinks to Library and Format objects, users can also click on Diff menu to compare the difference between two data libraries, and click Index menu to search a whole list of SAS variables alphabetically.

Fig. 5 The menu items in the header of main content frame of an HTML codebook

In an HTML codebook, dataset page has most complex structure. After header, dataset page comes a general description of the dataset and the hyperlink to sample data tables for the dataset, and then comes the summaries of three classes of variables within this dataset: individual variables, shared variables, and global variables. Each summary contains primary information about those categorized variables, such as variable name, type, and format currently associated. If a more detailed information for a particular variable is needed, a user can click on the variable name to open up another window that will present the detailed information about the variable such as distribution of the values, missing values it has, associated tables, and so on (See Fig. 6 below). The dataset page ends with a footer that contains the same hyperlinks as its header.



Variable *CAUSALIO*

Description			
Code associated with assessment by study site personnel of drug relationship to AE			
Basic Attributes			
Type	**Length**	**Format**	**Informat**
NUM	8		
Value Characteristics			
Dataset	**#Total Observations**	**#Distinct Values**	**#Missing Values**
PROT1234.AE	38	3	0

Associated Datasets Frequency of Values

Associated Datasets
PROT1234.AE PROT5678.AE

Go to top

Frequency of Values

Distinct Value(Total=3)	Count
1	28
2	8
3	2

Fig. 6 An HTML page for information about variable object

The HTML files for an HTML codebook are stored in several different sub-directories within a common root directory. This directory structure is designed to reflect the way in which HTML codebook files are typically organized for search and navigation. All directories can have nested sub directories and/or files.

The content structure and typography of the HTML codebook is defined by a collection of HTML templates. An 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 custom tags. The major advantage of using HTML template is to separate HTML presentation from data and codes that generate the HTML page. An HTML template normally corresponds to a navigational object in the HTML codebook navigational model. There are several HTML template 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 the HTML codebook for clinical datasets.

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. An HTML template for a navigational object is consisted of HTML and these new tags. Below is an example of HTML template for library Index access element in the HTML codebook navigational structure model.

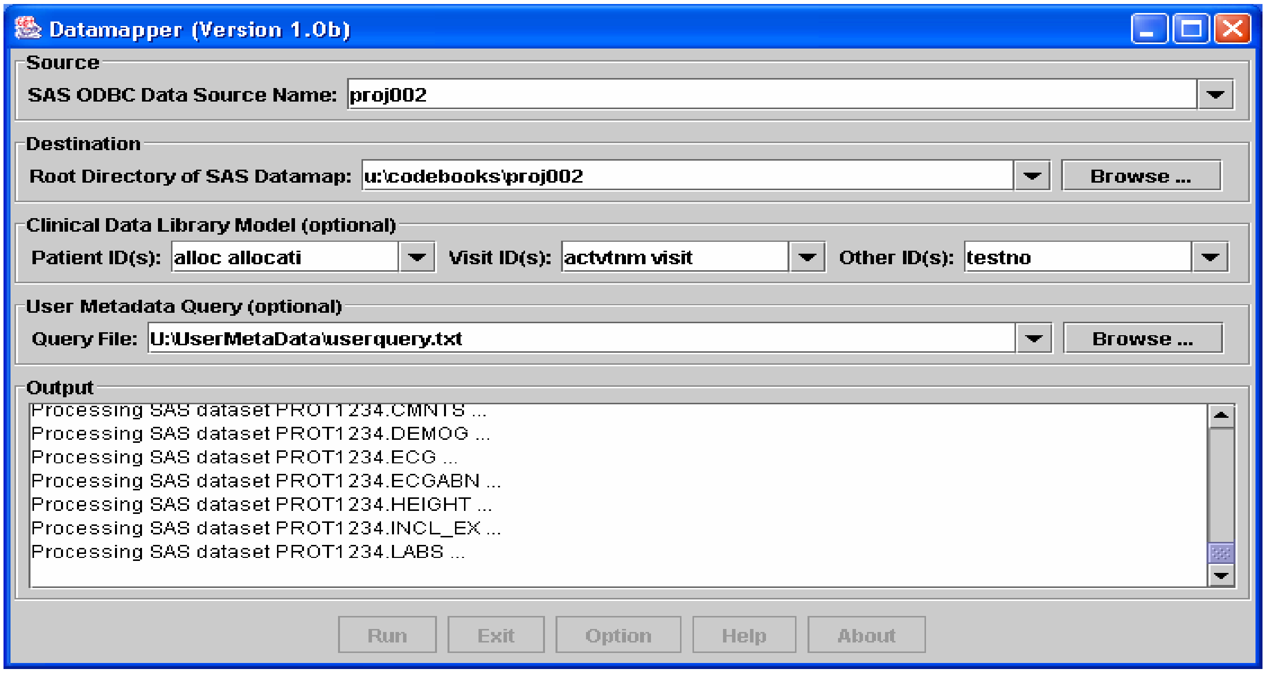
```
<html>
<head><title> Library Index </title></head>

<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 the HTML codebook generator called Datamapper uses the HTML. templates to generate HTML codebooks from given SAS ODBC data sources.

DATAMAPPER: AN HTML CODEBOOK GENERATOR FOR CLINICAL DATASETS

Datamapper is a very convenient tool for users like SAS programmers and statisticians to create standard HTML codebooks from clinical datasets with just a few clicks. The automated generation of the HTML codebook is critical, because (1) many SAS programmers and statisticians, often don't have additional overhead to service codebook requirements, not to mention of composing a comprehensive HTML codebook; (2) There is little or no margin to increase staff costs to fund the compiling of the HTML codebook for project-based clinical trial studies. Datamapper, written in Java, generates the HTML codebook from a given SAS ODBC data source with a set of predefined HTML templates derived from HTML codebook model described in previous sections. It has very simple user-friendly interface (See Figure 7 below). Zipped in one Jar file with size less than 150k, Datamapper can be launched locally or remotely when it is stored in a web server. It works with all SAS versions that have a SAS ODBC driver.


Datamapper (Version 1.0b)
Source
Destination
Browse ...
Clinical Data Library Model (optional)
User Metadata Query (optional)
Browse ...
Output
Run
Exit
Option
Help
About

SAS ODBC Data Source Name: proj002

Root Directory of SAS Datamap: u:\codebooks\proj002

Patient ID(s): alloc allocati Visit ID(s): actvtnm visit Other ID(s): testno

Query File: U:\UserMetaData\userquery.txt

```
Processing SAS dataset PROT1234.CMNTS ...
Processing SAS dataset PROT1234.DEMOG ...
Processing SAS dataset PROT1234.ECG ...
Processing SAS dataset PROT1234.ECGABN ...
Processing SAS dataset PROT1234.HEIGHT ...
Processing SAS dataset PROT1234.INCL_EX ...
Processing SAS dataset PROT1234.LABS ...
```

Fig. 7 Screenshot of Datamapper user interface

Datamapper takes two required inputs: the first is the SAS ODBC data source name (DSN), and the second is the folder that stores the datamap to be generated. Since a SAS ODBC data source can register multiple SAS libraries, Datamapper can be used to create HTML codebooks for any number of data libraries of interest. If users can provide optional clinical trial specific information, such as variables for patient ID, and visit ID, more domain-related meta-information such as clinical dataset type and patient visit flowchart will be extracted into the generated HTML codebook. Besides, it can generate HTML pages for user-written metadata queries about the clinical data.

Datamapper has a following structure illustrated in Fig. 8.

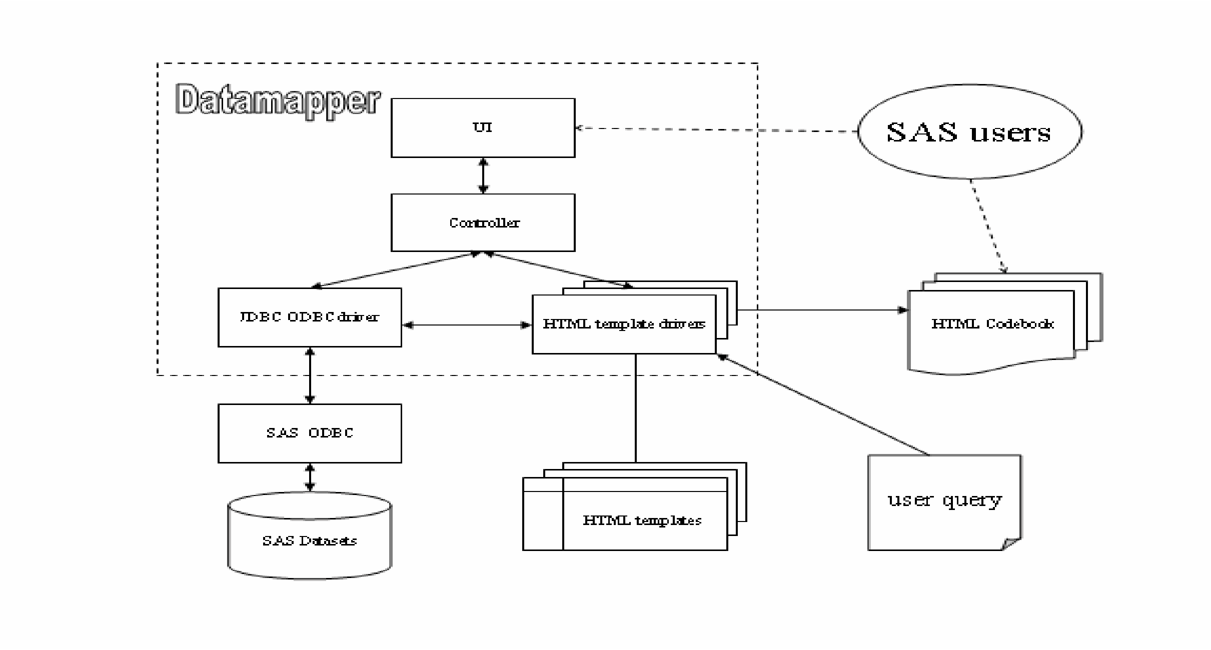


Fig. 8 Architecture of the Datamapper

As seen from Figure 8, Datamapper is consisted of Controller, HTML template drivers, JDBC-ODBC driver which access SAS data source, and UI module. The role of Controller is to take user inputs and hand over the task of generating HTML pages to HTML template drivers. An HTML template driver is a Java class that dynamically binds an HTML template with specified metadata that are retrieved from SAS ODBC data source through Java JDBC-ODBC driver and generate the corresponding HTML pages (for accessing to SAS ODBC functionality, please see [5] for more details). For example, below is a Java driver class for the HTML template of the library Index access element mentioned in this section.

```
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 user library names
    SQLExecutor se1 = new SQLExecutor(con, "select distinct libname from " +
     "dictionary.tables where libname not in " +
     " ('MAPS', 'SASHELP', 'SASUSER', ‘SASADMIN') order by 1");
     ResultSet rs = se1.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 the HTML codebooks due to the separation of representation and typography from the contents of codebooks.

CONCLUSIONS AND FUTURE WORK

This paper describes an HTML codebook model for clinical trial data and the Datamapper tool that automatically generates the HTML codebook from a given SAS data source with a set of HTML templates based on the model. The tool provides a convenient mechanism for both SAS programmers and statisticians to obtain standardized HTML codebooks quickly and easily, and browse them the way they surf Web without requiring knowledge of mundane details about clinical SAS datasets and additional metadata 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 clinical trial data source for different type of users, and the ability to integrate the HTML codebook with SAS source code HTML documents.

REFERENCES

1. Lei Zhang, "Datamapper: A Documentation Generator for SAS Metadata”, Annual Conference of Pharmaceutical Industry SAS Users Group 2003 (PharmaSUG 2003), Miami, Florida
2. The Object-Oriented Hypermedia Design Model (OOHDM) Home Page. http://www.telemidia.puc-rio.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-tmpl-java.sourceforge.net
5. Lei Zhang and Tianshu Li, “Using SAS ODBC with Java”, Proceedings of Northeast SAS Users Group 15th Annual Conference 2002 (NESUG 15), Buffalo, New York

ACKNOWLEDGMENTS

The author wish to thank Izabella Peszek, John Troxell, Elaine Czarnecki, Snow Fu and many other colleagues for their advice and support during the preparation of this paper and in the development of Datamapper application.

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 07065-0900
Work Phone: (732)-594-9865
Fax: (732)-594-6075
Email: Lei_Zhang4@Merck.com