Clinical Data Warehouse Functionality
Peter Villiers, SAS Institute Inc., Cary, NC

ABSTRACT
Data Warehousing is a buzz-phrase that has taken the information systems world by storm. It seems that in every industry publication, there are articles explaining how this relates to financial and marketing data, yet there is very little information about how the concept of data warehousing applies to the research and development side of the biomedical industry. This paper aims to review some information presented at PharmaSUG 1998 in the paper “The Clinical Data Warehouse”; detail how this information has been acted upon; outline the lessons learned and finally specify areas of new functionality that have been identified to further aid in the clinical research process.

This paper is aimed at anyone interested in obtaining a general introduction to clinical data warehousing functionality. The paper will describe concepts and ideas but will not go into details of implementation.

CLINICAL DATA AND TRANSFORMATIONS
Prior to explaining how data warehousing concepts can be applied to the clinical data environment, it is important to briefly define the scope of the clinical data environment. Figure 1 below shows the core data flow that clinical data warehousing functionality is aimed at.

In addition to these issues, the advent of collaborative development between, and mergers of drug development companies might lead to the raw data being in different formats on different types of computers. This situation means that the raw data must be transformed into a format that is more conducive to review and analysis. This has traditionally been performed using the SAS® System, by writing DATA step programs.

The analysis data sets that are produced by performing the transformations should match corporate or drug development project standards where these are available. Regulatory requirements dictate that the transformation code used and the log listings from the transformation programs be kept to validate that the data was migrated between formats correctly.

Once the data from individual protocols are transformed, these are often rolled up to generate special interest data such as the Integrated Summary of Efficacy or the Integrated Safety Summary.

For a more detailed explanation of the topics covered here, please refer to the paper “The Clinical Data Warehouse” in the PharmaSUG 1998 Proceedings.

CLINICAL DATA WAREHOUSING 101
In his book “The Data Warehouse Toolkit”, Ralph Kimball defines a data warehouse as “a copy of transaction data specifically designed for query and analysis”. Applying this to the clinical domain, it can be inferred that the clinical data warehouse is a copy of the patient data specifically designed for query and analysis. A key word in the preceding phrase is “copy” because this has regulatory implications, which will be explained later in the paper under the scope of replication.

GOALS OF A DATA WAREHOUSE
Again per Ralph Kimball, the goals of a data warehouse are identified as:

- The data warehouse provides access to corporate or organizational data.
- The data in the warehouse is consistent.
- The data in the warehouse can be separated and combined by means of every possible measure in the business.
- The data warehouse is not just data, but also a set of tools to query, analyze, and present information.
- The data warehouse is the place where we publish used data.
- The quality of the data in the data warehouse is a driver of business reengineering.

It is important to understand how each of these can be applied to the clinical data environment.

The data warehouse provides access to corporate or organizational data. In terms of the clinical environment, the corporate and organizational data is stored in clinical data management systems (CDMS) and adverse event reporting systems. At a minimum, the warehouse should provide access to data from these sources.

The data in the warehouse is consistent. Once data is extracted from a CDMS; read from stand-alone files or merged with reference data such as lab normal ranges, it should be put into a consistent format. For example, every time a user accesses a variable called AECODE in tables for protocols that use a given compound, the user should know that AECODE is defined the same and has the same meaning in each table that it appears.

The data in the warehouse can be separated and combined by means of every possible measure in the

Figure 1, The Clinical Data Environment
Data from clinical trials is normally entered and cleaned in a clinical data management system. These are either written in house or commercially marketed products. The clinical data in these systems is usually stored in a format designed specifically to aid with this job – a data entry oriented format. Very often, this is not the best format for analysis and review of the data, new variables might need to be derived from existing ones or reference data may need to be merged in with the raw data.

In addition to these issues, the advent of collaborative development between, and mergers of drug development companies might lead to the raw data being in different formats on different types of computers.

This situation means that the raw data must be transformed into a format that is more conducive to review and analysis. This has traditionally been performed using the SAS® System, by writing DATA step programs.

The analysis data sets that are produced by performing the transformations should match corporate or drug development project standards where these are available. Regulatory requirements dictate that the transformation code used and the log listings from the transformation programs be kept to validate that the data was migrated between formats correctly.

Once the data from individual protocols are transformed, these are often rolled up to generate special interest data such as the Integrated Summary of Efficacy or the Integrated Safety Summary.

For a more detailed explanation of the topics covered here, please refer to the paper “The Clinical Data Warehouse” in the PharmaSUG 1998 Proceedings.
business. Once the data has been transformed into an analysis ready format, the user should be able to merge and summarize the data in many ways.

The data warehouse is not just data, but also a set of tools to query, analyze, and present information. The final transformation of data is not the end point of this process for the data itself. It may be the goal of the programmer performing the transformation but other personnel will use this data: statisticians, medical reviewers, medical writers, regulatory affairs, etc. There should be tools available further downstream for these people to access the analysis data that has been generated using the warehouse.

The data warehouse is the place where we publish used data. Essentially, this means that any subsequent analysis or conclusions that are drawn from the data should be made using the transformed, consistent analysis data. Put another way – the analysis data is that which is used to make decisions.

The quality of the data in the data warehouse is a driver of business reengineering. This point is sometimes seen as a little controversial. The most basic explanation is that if a given transformation is very difficult to perform for every protocol that is controversial. The most basic explanation is that if a given transformation is very difficult to perform for every protocol that is added to the warehouse, this may be the driver [pain] that is required to change the input format into something that is more easily transformed and used.

THE PH.DATWARE™ SYSTEM

In 1999, PharmaHealth Technologies™, a business unit of SAS Institute Inc, introduced the PH.DataWare System to tackle the core functionality addressed in the previous section. The aim was to provide a controlled environment where the data from CDMS systems could be accessed and transformed into analysis data ready for use with clinical data review systems. This wasn’t a new concept; programmers had been doing this for years using the SAS System. What was new was the concept of a controlled environment aimed at supporting the entire process by capturing the information [metadata] about what was happening in the transformation process. This information included but was not limited to:

- Where is the raw data stored?
- What is the format of the analysis data?
- Which raw data sets are used as input to a program to generate the analysis data sets.
- What are the transformations that a data element goes through to become analysis data?
- Who is responsible for this protocol, data set, program, etc?
- What reports are created from an analysis data set?

By collecting this type of information [metadata], it was possible to add extra functionality into the transformation process:

Automatically capture and store the source code and log listings when data set transformations are run. A very important issue in today’s regulatory environments is being able to show what has happened to the data from the point the investigator notes it on the case report form. This includes everything that happens to the data after it has been cleaned in the CDMS. By removing the need for the programmer to store these and pass them on, this reduces the risk of incorrect information being given to the regulatory authority.

Automatic generation of documentation from stored metadata. As a result of registering the metadata, documentation can be generated quickly and efficiently relating to the structure of the analysis data sets; the raw data sets used as input into the process to generate them; the date of the transformation process and also the source code used and the log listing produced when the program was run. In addition, this can be produced as HTML so personnel can view it using a web browser.

Calculation of dependencies between transformations. If analysis table DEMOG is used as input in the transformation process for analysis table AE, then AE is said to be dependent on DEMOG. This is because if AE’s transformation process is run before DEMOG’s, the resultant information in AE that came from DEMOG might not be correct once DEMOG’s transformation process has run. Using the captured metadata, it is possible to automatically determine these dependencies and therefore ensure that the analysis tables in a given protocol are regenerated in the correct order. A screen shot of this is shown in Figure 2 below.

Figure 2, Job Dependencies

It is also possible to “tack on” the production of reports and the automatic documentation mentioned previously into this process. This enables the user to regenerate the analysis data for a protocol, produce the documentation and subsequently run reports from the new data in one operation.

Determination of impact analysis. Whereas the dependencies concentrate on which transformations need to be carried out in order to produce a set of tables. Impact analysis concentrates on which tables (and reports) need to be rerun if the data for a given source table changes. An example of this might be that all the transformations for a protocol have been registered and these data sets have been used to generate summary data sets across the drug development program. It is discovered through an audit that a value in one of the raw data sets is incorrect and needs to be updated in the CDMS. The impact analysis functionality can use the metadata to determine which tables are generated from that newly updated raw data set (and which tables are generated from them, and so on) and can re-run all the transformation processes in one operation. A screen shot showing the impact analysis is shown in Figure 3.

Figure 3, Impact Analysis

As can be seen from functionality described above, the clinical data warehouse is not just a place to store data. It is a central repository of information that can be used to reduce the risk to an organization. It does this by capturing the actual process that was used to manage and transform the data that is more than the generic information specified in standard operating procedures.
FURTHER FUNCTIONALITY
Since PH.DataWare has been available, the industry has provided a great deal of interesting and valuable feedback relating to the additional functionality that would be required of a clinical data warehouse. Much of this is based on usage of the system, however, some of the feedback is a result of the perceived needs of regulatory authorities. As a consequence, there are other requirements that would broaden the base functionality of a warehouse. A few of these are described in the following paragraphs.

STANDARDIZATION
While allowing analysis table definitions to be copied and providing search facilities through the metadata, there has been a call for “one version of the truth” when defining the columns of analysis tables. To achieve this and other goals, the “lexicon concept” was created. In its most basic form the lexicon is the column definitions that may be used in any analysis table created by the warehouse. When defining the columns in a given analysis table, the user can select from previously defined columns or add a new one if it does not exist. As can be seen from Figure 4 below, different tables will share the same column definition, and when the warehouse performs the transformation to create the analysis tables, the user can be assured that the columns match the standards that are defined.

In addition, if standards change or a modification must be made to the label for a column, it will only need to be made in one place, and because of the metadata, the warehousing system will be able to inform the user which analysis tables are affected and therefore, which transformation programs need to be run again to ensure that the data set accurately reflects the desired structure.

In addition, if standards change or a modification must be made to the label for a column, it will only need to be made in one place, and because of the metadata, the warehousing system will be able to inform the user which analysis tables are affected and therefore, which transformation programs need to be run again to ensure that the data set accurately reflects the desired structure.

1 to 1 Mappings. These are basic mappings where only the name, length, label, etc need to be changed on a variable.

SAS Snippets. These are pieces of code that are written to create the analysis column required. The final product of these is a temporary data set containing the required keys and the actual analysis column.

1 to 1 Mappings

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>char</td>
<td>10</td>
</tr>
<tr>
<td>Patient</td>
<td>char</td>
<td>10</td>
</tr>
<tr>
<td>Visit</td>
<td>num</td>
<td>8</td>
</tr>
<tr>
<td>Aecode</td>
<td>char</td>
<td>10</td>
</tr>
<tr>
<td>Bdate</td>
<td>date</td>
<td>8</td>
</tr>
<tr>
<td>Age</td>
<td>num</td>
<td>8</td>
</tr>
</tbody>
</table>

SAS program code if birthdate in raw data was character

Figure 5, Column Definitions and Code Snippets
As can be seen from Figure 5, most of the columns have 1 to 1 mappings; however, AGE has two code snippets associated with it. The first to convert a character variable representing the date of birth to an age. The second to convert a numeric variable representing date of birth to an age.

When these code snippets are added to the lexicon, they are validated and metadata relating to when, and who registered the code, and who performed the validation is captured.

From a user’s point of view, once an analysis table is defined and columns are added to this from the lexicon, the user can then select the appropriate transformation to use from code that has already been registered and validated. In addition, because metadata exists about which transformations were used in previously defined protocols/tables, the user can easily be presented with this information to aid in their selection.

By splitting the code up in this way, it enables more frequent reuse of previously validated code. This appears to be a cleaner alternative than taking a large DATA step program to create an entire data set, modifying it slightly to meet current needs an then being required to revalidate the entire DATA step, even the lines of code that were not modified.

Returning to the challenge of producing column level documentation – this could be produced automatically now because of the captured metadata. Not only would this be possible but should a reviewer want to inspect the transformation logic for a given analysis column, it is much easier to locate and validate just this section, rather than try to unhook this logic form a potentially large DATA step that performs many transformations.

MORE CODE GENERATION
Given that the analysis table is defined and the column transformations are selected, when a user specifies that the data in the table be refreshed the system must produce the...
The source code section (over) gives an idea of the code that would be produced. The metadata collected about the column definitions and code segments would allow the system to essentially write the administrative comments that identify each section of code:

- Table information at the top of the program identifying it and the protocol that it belongs to.
- Column definition information
- Column transformation information followed by the code segment itself.
- A merge section at the end that pulls the individual working data sets together. This would be completely generated by the system based on table metadata.

```sas
/***********************************************************/
data analysis.demo;
  length protocol $10
  patient $10
  bdate $8
  age $8;
  format bdate date9.;
  label protocol = 'Protocol number'
    patient = 'Patient number'
    bdate = 'Date of birth'
    age = 'The age of the patient in years';
  merge tmp0001 tmp0002 tmp0003;
  by patient protocol;
run;
```

**LINKS TO EXTERNAL DOCUMENTS**

In line with the warehouse being a central repository on information, not just data, the warehouse should allow users to link to external documents. These could include the actual protocol, coding standards for the organization, correspondence with the appropriate regulatory authorities, or even resumes of personnel working on the project. It should allow these documents to be linked with almost any information that the warehouse itself stores.

Organizations may store this “external” information in documents on disk, intranets or document management systems. In the latter case, the warehouse should allow the user to be able to point into the document management system at the appropriate document.

By enabling this, the organization would always have the current information at hand to refer to during the course of work and know which information was used retrospectively.

**REPLICATION**

An attribute of today’s biomedical industry is an increase in distributed development. It is more common that not all the work is centered in one location but around the globe. For instance, a company might have the raw data, transformations and analysis data stored and performed in the USA, whilst interested parties in the organization might need access to the data and metadata in Great Britain. Of course, organizations today have a number of options to solve this problem.

- **Global networks.** The data and metadata stays physically located in the USA and the system is accessed from the UK via the same application.
- **Corporate Intranets.** The information is made accessible via the web – no need for the system to run in Great Britain.

Both of these suffer from the same problem if a lot of data needs to be brought over the international network on a frequent basis – degraded network performance. In this case, it might be a better idea to provide replication services to allow the data and/or metadata to be copied in a read-only format from the warehouse in the USA to a warehouse in Great Britain. The data and metadata are only transferred across the Atlantic once then are accessed “locally” from that point.

Ideally the warehouse can be instructed to do this automatically at given intervals or when the data/metadata is updated.

**THE FUTURE**

One word – Genomics. Accessing, standardizing, storing, merging the acquired genomic data back into the traditional clinical data and mining the resulting combined data.

Not only will the industry have the current challenges but it will have new ones relating to the special nature of genetic information. New tools will be required to aid in the management and exploitation of this.
PharmaHealth Technologies is developing a web-ready, platform architecture to support this integrated approach and new viewers to explore this fascinating data. The tools will combine the functionality identified here with new algorithms specifically for genomic data.

CONCLUSION
As can be seen from the above information, a great deal has already been identified and accomplished in the area of clinical data warehousing. As the industry has become more attuned to the advantages of warehousing data, more demands have been made of the warehouse that has broadened its use.

The author believes this will continue in the future and the idea of a clinical data warehouse will be overtaken by the idea of a product development warehouse containing (or linking to) all information pertinent to a molecule.

REFERENCES
“The Clinical Data Warehouse”, Peter Villiers, PharmaSUG 1998 Proceedings
“The Data Warehouse Toolkit”, Ralph Kimball, Wiley Computer Publishing

CONTACT INFORMATION
Peter Villiers is the project manager and lead architect for biomedical data warehousing at SAS Institute Inc. He has a BSc. (hons) degree in Computer Science from Kingston Upon Thames University, Great Britain.

Your comments and questions are valued and encouraged. Contact the author at:
Peter Villiers
SAS Institute Inc
SAS Campus Drive
Cary, NC, 27513, USA
Work Phone: +1 (919) 677-8000
Fax: +1 (919) 677-4444
Email: peter.villiers@sas.com

SAS, PharmaHealth Technologies, SAS PharmaTechnology Process, and PH.DataWare System are registered trademarks or trademarks of SAS Institute Inc in the USA and other countries. ® indicates USA registration.

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