Paper 115

Data Warehousing in Pharmaceuticals and Healthcare: An Industry Perspective

M. Kumar Sagar
The Sagar Group, Inc., Framingham, MA

Himanshu Raval
Pfizer, Inc., New York City, NY

ABSTRACT

This paper aims to describe the data warehousing process for the Pharmaceutical and the Healthcare industries. We will first examine the reasons and needs for the data warehouses and the expected benefits. Based on this, a general process for data warehousing emerges that utilizes a host of tools and techniques. These are then illustrated by two case studies as follows:

(1) A Clinical Trials Data Warehouse for the Pharmaceuticals and Biotech companies,
(2) A Utilization and Claims Warehouse for the Drug Manufacturers and the HMOs.

Each of these warehouses has different design philosophies, objectives and utilization. Each of these case study warehouses uses a different set of tools for populating the warehouse, data transformation and data extraction. While discussing these case studies, we will also delve into the subjects of:

(1) User Requirements
(2) Design Specifications
(3) Validation and Testing
(4) Documentation
(5) User Training and Setup

INTRODUCTION

In the current post-industrial information society, data is one of the most valuable resources. However, data is not useful unless it can be processed and turned into information that allows enterprises a competitive advantage. Moreover, our ever-more efficient information systems are collecting ever-increasing amounts of data. Further, while data may be “out there”, it may be in many disparate forms and formats, essentially making it unusable. So, what are we to do with these mountains and islands of data to help us test hypotheses, derive conclusions, and identify trends and opportunities?

It is in this context that data warehousing can help us turn data into information amenable to analysis, data mining, trend identification, and respond to these trends in a beneficial way. Thus, it allows enterprises to cope with the well-known phenomenon of information overload. To be able to fulfill this agenda:

(1) Data has to be accessible,
(2) Data has to be current,
(3) Data has to be flexible, and
(4) Data has to be understandable.

Furthermore, to be successful, a data-warehousing project has to be aligned with business objectives to assure that the project is successful and delivers return on investment.

ALIGNING TECHNOLOGY AND BUSINESS

To be successful, a data warehouse needs to aligned on the following 3 criteria:

1. **Business Criteria**
   1.1. Critical Success factors: Why do data warehousing?
      1.1.1. What is the problem to be solved?
      1.1.2. What responsibilities/goals are directly related to the problem at hand?

1.1.3. What are the critical success factors?
1.1.4. Which organizational components are best suited to solve this problem?
1.1.5. What is the audience? Who will it benefit? Why?

1.2. Quantify Benefits
Expected benefits must be quantified for two reasons: to be able to measure the success of the project and to justify the expenditure.
1.2.1. Cost Avoidance Benefits
This simply means savings in cost due to not having to do tasks needed earlier such as report distribution, reduction in user requests, etc.
1.2.2. Efficiency Gains
These primarily result from increased productivity because of reduction in end-less loops and continuity of the task.
1.2.3. Warehouse Savings
When warehouse-based decisions are better quality, they can result in reduction in cost.

1.3. Product Integration Issues
Due to lack of maturity and continuous changes in the technology arena, there exists a proliferation of tools and technologies out there in the market. This obviously leads to a risk of how well these products will integrate, what common platforms do they support, can they all use a common metadata, etc.

1.4. Cultural Issues
Since data warehousing requires pooling together of resources from across departments and sharing, this may mean loss of control of data. These impediments and mindsets need to be dealt with to make the warehouse project a success.

1.5. Standardization Issues
Since a warehouse may involve multiple tools from multiple vendors, they not only need to integrate well together; they should also integrate well with the existing technology infrastructure. An oft-used strategy is to align with the largest or dominant players in the arena.

1.6. User Training Issues
To assure that data warehouse is used in appropriate ways and efficiency gains are maximized, users should be trained in navigating the data schema/model, querying, reducing runaway queries and reporting.

2. Process Criteria
2.1. Scope
This essentially deals with questions such as thinking about a single repository to store all the data rather than many smaller ones. Yes, if it can be realized, benefits could be significant but it may not be realistic due to cooperation and integration required among many enterprise units, standardization of terms and definitions, and the cost of the project. In contrast, a departmental data mart approach will be much smaller in scope, much quicker to achieve and less costly. However, such a data mart may be developed in vacuum, leading to yet another island of data.

2.2. Implementation Approaches
2.2.1. Integrated Package
2.2.1.1. Faster
2.2.1.2. Reduced Cost
2.2.1.3. Integrated
2.2.1.4. Low Technology Risk
2.2.1.5. Less Flexibility
2.2.1.6. Compromises in Functionality and Performance
2.2.1.7. Easier Upgrades
2.2.1.8. Single Vendor
2.2.2. Best of breed Solution
2.2.2.1. Slower
2.2.2.2. Higher Cost

- 2.2.2.3. Integration is hard
- 2.2.2.4. Higher Technology Risk
- 2.2.2.5. Higher Flexibility
- 2.2.2.6. Compromises in technical architecture
- 2.2.2.7. Complex Upgrades
- 2.2.2.8. Multiple Vendors

2.2.3. In-house Development

- 2.2.3.1. Slower
- 2.2.3.2. Highest Cost
- 2.2.3.3. Expertise may not be available
- 2.2.3.4. May over-burden IT Department
- 2.2.3.5. Single/Multiple Vendors

2.3. Top-down or Bottom-up?

This depends on whether the focus is on solving the short-term issues vs. the long-term strategic issues. Thus, an organization aiming to solve departmental level issues may decide that a data mart is appropriate whereas an organization facing issues such as multiplicity of data being collected in multiple places and lack of common definitions just may decide that only a top-down approach will work. In this instance, setting up an enterprise level architecture and standards and then constructing warehouses/marts that conform to it may be the only answer. While the top-down approach may yield the best long-term results, it may be very difficult, expensive, and time consuming. In contrast, the bottom-up approach may just replicate the current “islands of data” and create issues of lack of scalability, lack of standardization, and different interpretations.

3. Technology Criteria

3.1. Scalability

Scalability is defined as the ability of a system to grow as users demand more, as data volume grows, as user population increases, and as the analyses complexity grows. Since it is very difficult to estimate the actual system loads as well as the actual uses it will be put to, scalability in a data warehouse is a must.

3.2. Manageability

Data warehouse product set must be able to support and manage solutions to a number of issues such as:

- 3.2.1. Managing the extraction and transformation of data – frequency, error handling
- 3.2.2. Reporting Flexibility
- 3.2.3. System Administration
- 3.2.4. User Administration
- 3.2.5. Backups and restores

3.3. Performance

Performance of the data warehouse system can be the major factor dictating whether it will be a success or a failure. If people can not use it effectively and efficiently and the response time is not adequate, users will not use it.

To address performance issues, both hardware architecture (SMP vs. MPP) and database architecture must be examined, adjusted, and tuned.

3.4. Flexibility

A data warehouse should be flexible enough to respond to changing technology, user demands and marketplace. It should allow for rapid responses to the changes in the technology itself.

3.5. Ease/Speed of Implementation

A tightly integrated development and maintenance tool set will obviously help in the success of the project by allowing developers to respond to user demands and changing conditions in a timely manner. Important attributes of such a tool set are:

- 3.5.1. Interface – GUI vs. command line
- 3.5.2. Single-step metadata generation vs. multiple steps

3.5.3. Version Control
3.5.4. Plug-and-Play capabilities

3.6. Tool Integration
This deals with how well different components interact. For example, metadata integration will assure that metadata will always be synchronized without bringing down the system. Furthermore, tools must also work with the existing hardware and network architectures without too much custom code.

IMPLEMENTATION TACTICS

1. Project Planning

1.1. Management Sponsorship:
Sponsorship is a very crucial element and the level of support should be commensurate with the scope of the project. In other words, an enterprise level project should be sponsored by the President/CEO.

1.2. User – IT Partnership: Since users may not know very much about their data as well as analysis requirements due to the newness of the DW technology, it is critical to form a team where both developers and end-users work hand-in-hand to take the project through multiple prototypes and iterative development. Thus, it is a good idea for applying RAD and JAD.

1.3. Success Metrics: A clear metric will allow the organization to check whether the project is successful and what can be done to address the deficiencies.

1.4. Project Plan: A project plan detailing the objectives, approach, strategy, timeframe, resources and deliverables is a must.

1.5. Resource Identification and Planning: Identifying resources and staffing the project – whether in-house, additional hiring or vendor supplied – can affect the timely and successful implementation of the DW project to a great extent.

1.6. Communication and Feedback: An enterprise or any of its sub-units should be communicated to clearly as to what the project will deliver and what the users can expect and how to get the most out of it.

1.7. Service Providers: Data Warehousing is a service business. Thus identifying vendors to address the areas where the enterprise lacks the expertise should be carried out very early on and appropriate resources should be secured.

2. Architecture
Architecture is both defining the components and their interactions. It should be scalable, flexible, extensible, manageable and robust. The choices could range from a tightly integrated product suites to individual blocks/products that IT would integrate into a single end-to-end solution.

3. System Design
It should be open and well understood with well-defined component interfaces. Standards can help the system be more extensible and flexible. User input should drive the system design and care must be taken to avoid the "tunnel syndrome"

4. System Development
Using RAD and JAD can help the users understand the benefits of DW and how it can help them be more productive. Tools should be provided to update the metadata, the "heart of the DW", so as to reduce the errors and the complexity. An administration tool can be of enormous importance and the system must provide for it. A solution put forward by the Metadata council is to pass all metadata through a "metahub" that standardizes it.

5. **Piloting and Prototyping**
   Whenever the technology is foreign to the organization, the technology is understood but the application is not, or that the cost benefits are not understood; it is a great time to pilot and prototype it to shed more light.

6. **Deployment and Training**
   It should consist of a rollout plan, a training plan and a support plan.

7. **Managing the Warehouse**
   Provision must be made to manage the warehouse so as to be able to improve it and capitalize on the investment. Managing consists of:
   7.1 Timely transfer of data files
   7.2 Data extraction processes
   7.3 Data Cleansing process
   7.4 Metadata administration
   7.5 User Administration
   7.6 Schema optimization and modification
   7.7 Problem resolution

Given this overview of the data warehouse theory, now let's examine some real life cases.

CASE STUDY 1:
A Clinical Data Repository

Introduction

In a typical clinical trial, especially large and/or multicenter ones, there are many sources of data, including electronic data transfers from sites, central labs and CROs. Moreover, with multiple trials going on across many projects for the same compound/drug, the issue becomes how to manage all this data and at the same time, not repeat data collection. Thus, to summarize briefly, the data is received in house, is cleaned up, required transformation routines are applied to massage and reformat the data, stored in an appropriate data format, and reports/analyses are run. If a central repository is available, it may result in:

- Faster and easier data cleanup
- Faster reporting
- Quick Integrated Safety and Efficacy Summaries and other Analyses
- Faster time to market

The Company

The company, a large multi-national pharmaceutical giant, has a plethora of clinical trials for a number of drug projects. Moreover, data collection and analyses operations are spread across the world, making it harder to enforce data standards. Even harder to enforce was the programming and validation standards that are required of pharmaceutical companies.

Clinical trials are run in scores of countries and the collected data is needed by clinicians and statisticians from every site for analysis and product defense.

Why Data Warehousing

A data warehouse is primarily an operational middle ground where a large number of disparate and incompatible "legacy systems" are tied to an equally diverse collection of end-user platforms.

Legacy systems, in this case, comprised of a hodgepodge of assorted hardware, software, and operational systems accumulated over many decades. By their very nature, they were quite incompatible with one another.

The end-user platforms were equally diverse. Users had Windows, OS/2, and Unix workstations. Some were even using dumb terminals. All these end-user workstations were tied to one or more or none of the legacy systems. And according to Murphy's Law, this is when things went from bad to worse. It was quickly realized that certain people using dumb terminals knew a lot about the system they were using but would quickly bog down when confronted with a PC with a graphical interface. Dec/VMS programmers merely refused to even consider looking at an IBM mainframe and IBM/JCL programmers held everybody else in contempt. The Unix programmers were, of course, in their own alternate universe.

The data warehousing concept gained ground when it was first proposed and all users were immediately trained in the use of a Windows based PC. The single (at first) original target of the whole operation was that everybody should be able to access whatever it is they wished from their PC. Upto this point, the only picture that could even be remotely considered rosy was that everyone needed access to SAS datasets which happens to be the *de facto* standard for pharmaceutical companies. Only, they were on a number of different platforms and therefore in different formats. And thus a decision was made which drove a lot of people insane or almost insane.

How was it done

Management Sponsorship

A proposal was made and the project was sponsored by the President of the Biometrics division. Most users in this case knew about the data and the analysis requirements but did not know the means of actually implementing anything technical. The corporate IT group was persuaded to partner with the Biometrics group much to the detriment of both parties.

External consultants with experience in implementing data warehouse technologies and software project planning were brought on board and the project began in earnest.

Project Plan

A detailed project plan was developed which encompassed initially all the technical parts of the whole project. Another project plan actually dealt with the implementation of the data warehousing project itself. Various milestones (achievable) were identified and incorporated into the project plan.

Identification of Resources

This turned out to be the toughest part of the whole project. Eventually this was accomplished; the Unix people were dragged out of their alternate universe; the VMS people agreed to look in a certain direction and the IBM/JCL people were made to realize that everyone else was also human. External vendors/consultants with the help of the division VP were required to do this.

Architecture

The existing architecture comprised of IBM and DEC mainframes. Most users had a PC on their desktop though with different operating systems. A Novell NetWare LAN was in place and so was a Windows NT network. TCP/IP was the standard protocol used throughout the enterprise.

System Design

A conscious decision was made to use an “open system” to actually hold the data warehouse. Unix running on Sun ES6500s was the platform of choice.

Most of the data was already in SAS datasets but data from clinical trials conducted in various other countries also occasionally showed up in Excel spreadsheets and Oracle export files. This data warehouse was also a very specialized application in that it was geared towards only the biometrics divisions within the enterprise.

User input in the actual design was varied and extremely interesting. The SAS programmers wanted the data to be extracted straight from an Oracle database where it was entered and cleaned. They also wanted the ability to write their own programs in the same way they had before. The proponents of the “validated system” wanted the data warehouse to actually be a “validated process environment”. Some clinicians wanted to look at the paper output while some of the more liberal ones wanted the ability to browse the data online.

Most programmers, clinicians, and statisticians in the Phase IV group had their own requirements which needed to be addressed. This group’s needs were not geared towards an NDA (New Drug Application) but more towards surveillance of the current market for the drugs. Certain statisticians/clinicians in the group were more geared towards answering questions of efficacy from the FDA and other drug users.

It should be noted here that though a large number of clinical trials are run to check the efficacy and safety of the drug, only a certain number of these trials are actually used in the actual submission process. The rest of the trial data is still available and could be used to do a trend analysis and also for testing other indications for the same drug. People in the Phase IV group wanted access to this data as well.

Around this time, the IT group bailed out promising us that while they would most certainly take care of the Unix systems and the network; we were on our own as far as data warehousing goes.

Because of these reasons and needs, the PH/Dataware was selected as the software of choice. This also satisfied the "validation" group as the Clinical Data Warehouse (as we called it) was developed by the SAS institute.

System Development

The system development was handled by the SAS Institute PharmaHealth Technology group. After gathering the requirements, a prototype was demonstrated and enhancements were requested. This iterative process is still continuing. The process for managing the warehouse is being implemented and certain users are being trained for this task.

The Changes

The introduction of the data warehouse ended up introducing a couple of high-end Sun Solaris servers and Windows 95 on all users desktops. Strangely enough this was the only change in the hardware infrastructure.

In the arena of software, PH/Dataware system was introduced but people did not even perceive a change since it is entirely developed using the SAS System. Additional software tools were incorporated for reporting; querying, and data-mining. Most of these tools were also based on the SAS system. Quite a few of them were actually created in-house.
The social changes were immense. People actually started talking to each other especially about what they were doing and how they were doing it. A lot of in-house skills were suddenly brought to the forefront and even more software project management skills were learned during the course of this episode.

CASE STUDY 2:
A Utilization and Claims Warehouse

Introduction

In the arena of Health Insurance, millions of claims are generated everyday, including hospitals, physicians, and pharmacies. A number of clearing houses process these claims and route them to the appropriate payer. Thus, these clearing houses sit on a huge amount of data that runs in 10s of gigabytes everyday but other than routing them, no other insights are gleaned from these data. It was in this context that we were called in to help develop a data warehouse. If a central repository is available, it may result in:

- Drug Positioning Information
- Disease related Information
- Regional Preferences
- Market Penetration

The Company

The company, a small shop, is based on the east coast and gets data from the claims clearing houses and Pharmacy Benefit Managers (PBM) on a weekly basis. It processes the data and completes analysis requirements specified by the drug companies and the health insurance companies.

Why Data Warehousing

A data warehouse will allow seamless access to all the data and allow us to analyze it in a time series fashion. Briefly stated, following were the objectives:

- Supports data loads from a variety of sources such as HMOs, Pharmacy Claims Processors and metadata vendors
- can partition data in related subsets using predefined algorithms
- provides library functions for code reuse, maintenance, and development
- can integrate with SAS for data analysis

and reporting, using SAS/Access

- provides data security and management
- is easy to maintain, upgrade, and administer
- is scaleable, robust, and dependable
- can provide long-term strategic benefits in terms of information storage, retrieval, analysis, and presentation

Thus, such a data warehouse can provide invaluable intelligence in terms of:

- Drug Positioning Information
- Patient Population Characteristics
- Indications drug is being used for
- Prescribing Physician Characteristics
- Regional Preferences
- Prevalence of Diseases
- Preferred Drugs for Diseases
- Procedures being performed
- Disease related Information

How was it done

Sponsorship and Communications

The project was sponsored by the President of the company since he did see the benefits of the concept and was enthusiastically welcomed by everyone. Being a small company (10 employees), the communication of expected benefits was also very easy.

Project Plan

A detailed project plan was developed which encompassed all the technical parts of the whole project such as data extraction, data transformation, data cleansing, data storage, and reporting/querying. Another project plan actually dealt with the implementation of the data warehousing project itself. Various milestones (achievable) were identified and incorporated into the project plan.

Identification of Resources

Since the project was staffed exclusively by the outside consultants, this was the easiest part of the whole project.

Architecture

The existing architecture comprised of Sun ES4500 machine with a 1 terabyte RAID array and Windows95 PC on the desktops. TCP/IP was the standard protocol used.

In the old world, data was received on tapes and floppies. They were then stored on the RAID and SAS was used to convert them to SAS datasets, which were subsequently used in analysis and reporting.

A decision was made to receive data using EDI and stored on the RAID directly. This necessitated installation of dedicated lines to the claims vendors and PBMs.

System Design

Once the data was available on the RAID, various Oracle PL/SQL algorithms were used to read the data in, partition it, cleanse it, add value using decodes and formats, and then store them in the STAR schema of the data warehouse. Please see Table 1 for more information.

System Development

1. **Oracle DB Tuning**: While another vendor had already completed this task, we needed to revisit it to redefine, tune, and troubleshoot DB schema, DB parameters definitions, space allocation, and relational structures.
2. **Data Warehouse In-Table Load Programs**: This task required developing the load programs to load data in the warehouse In-Table from various sources. In-Table is the central table where all in-coming data is received and then segmented and dispatched on to the various production tables.
3. **Data Partitioning and Populating Production Tables Programs**: This was implemented in stages, with the first stage populating the production tables for one disease. This was then repeated for other diseases. It consisted of a number of Oracle PL/SQL programs to process the data loaded in the In-Table to partition it in segments and populate the production tables with these data.
4. **Loading Existing Data in Oracle**: This task was our test-bed for testing the programs as they were developed and was highly labor and time intensive.

5. **SAS/Access Installation**: SAS/Access software was installed and tuned to operate with the Oracle Data warehouse.
6. **SAS View Creation**: This step will involve creating SAS views for accessing the data from Oracle seamlessly. Each production table in the warehouse had a default view for all the variables in that tables as well as tailored views to select only specific variables and/or subsets of data.
7. **Front-end System Development**: Using Developer/2000, a front-end GUI system will be designed for easier management of data load and partitioning programs. This system was named *MetaManager*.

Metadata Management

MetaManager allowed the administrator to specify the source data (filename to be processed), target STAR tables, extraction algorithm(s) to be applied, and decode libraries to be used. Similarly, *MetaManager* allowed the administrator to update the decode and reference libraries as well as edit the extraction and cleansing routines. *MetaManager* also created SAS/Access views that would allow SAS programs to access the data warehouse seamlessly. Since Oracle, SAS and data warehouse were stored on the same machine and RAID, the amount of additional network traffic was minimal.

The Changes

The introduction of the data warehouse required people to lean new software packages such as Oracle and SAS/Access but management commitment and support ensured that appropriate people received the right training.

CONCLUSION

Based on our experience with this utility development process, here are some of the lessons we have learned:

1. Work closely with the end-users to identify the requirements and anticipate the likely uses of a data warehouse.
2. A well-defined project plan is a must for a data warehouse and it is very important to let a third party or vendors knowledgeable about your business manage it.
3. Keep the project simple and tightly controlled. Keeping it simple allows the project to remain manageable while tight control assures that end-user requirements and expectations do not run amok.
4. Keep programs simple and modularized. The more modularized they are, the easier it is to adapt them to ever changing needs.
5. Metadata management is one of the most critical aspect of a data-warehousing project.

REFERENCES

ACKNOWLEDGMENTS

We would like to thank Steve Ober, Anwer Khan, and Len for their invaluable assistance in letting us put our concepts to test.

AUTHOR CONTACT

M. Kumar Sagar
The Sagar Group, Inc.
37 Concord Street, Suite 6
Ashland, MA 01721
Phone: (508) 881-1480
Fax: (508) 881-1480
E-mail: sagar@sagargroup.com
URL: http://www.sagargroup.com

Please visit our site for more information and for an on-line, electronic version of this and other papers.

Himanshu A. Raval
Pfizer, Inc.
205 East 42nd Street
New York, NY 10017
Phone: (212) 573-7439
E-mail: ravalh@pfizer.com

Table 1
Case Study 2: A comparison of Old and New World

	Current Process	Proposed Process
1	Data received from various sources, various formats.	Same as current but using EDI.
2	Depending on the input file format, availability of personnel, and urgency; a software is chosen to be the pre-processor and another software is chosen to be the repository. It could be SAS, Access, Excel, Word, Wordpad or some other software. This plethora of software result in: data integrity issues and data inconsistencies.	Always loaded in Oracle. While Oracle is the repository, PL/SQL will be the pre-processor. Eliminates the problems associated with skill sets; data integrity and data inconsistencies.
3	This uploaded data is then processed and sometimes flags are created which are not always consistent. The data is also used to create: (1) reference datasets (2) analysis datasets (3) subset datasets OS and software limitations dictate how big these datasets/files can get and how much data it can hold (currently up to a maximum of 2 GB). Many of the reference tables have to be updated manually. This is accomplished by copying existing programs to a new location, modifying them, running them against the new data, and storing the final reference table in yet another place and/or with a different name. This not only creates a version management headache, it also reduces personnel effectiveness since it requires a greater effort to understand what programs were used in creating the data and what was the logic. This also wastes disk space, requires various people to contribute to the process at different stage and requires that everybody is aware of all the data characteristics such as variable names/types/formats/lengths, etc.	Data will always go through pre-processing and populating programs to assign appropriate flag values for easier subset generation and analysis. The data will pass through the "In-Table" and then channeled to various tables which may be either (1) longitudinal datasets (quarterly table) (2) analysis datasets (disease tables) (3) subset datasets (client datasets) Theoretically how big the tables can get is limitless, thus allowing a larger collection of both longitudinal as well as related data. In case of standardized algorithms, the same algorithms will process the data, whether it is first time or update. This will put a centralized data load and update control in place, thus streamlining the process and allowing greater specialization of people on various aspects of process. Standardized algorithms will also allow the reference tables to be updated automatically using programming techniques. Since the same algorithm will process the same data, this consistency will allow analysis programs in SAS to be highly standardized. This will reduce disk space requirements and less people intervention will be required. Analysts can be assured that the data characteristics will remain same so they can focus on the task of analysis rather than data scrubbing, data transformation, and program modifications to fit the new dataset.
4	Pre-processor Programs are currently written in SAS for some while many other type of data go the route of Wordpad -> Excel -> Access -> SAS. These software do not have the native interface for data processing from some other format so it takes longer to process and results are almost invariably different each time.	Pre-processor programs will be in PL/SQL and SQL, all supported by Oracle. Since these programs can be standardized and since they work only with Oracle, they will be highly efficient due to native support and results will be consistent.
5	NA	Populating programs will be written in PL/SQL and SQL that will channel data to the appropriate tables, create appropriate flags, and create summary data/table for administration purposes.