Challenges of Building Clinical Data Analysis Solutions

George W. Gray

Increasingly, owners of clinical information systems are turning to clinical data warehouses (CDWs) to store and to analyze their data. The CDW allows institutions to make better use of their clinical data that has been collected through its information systems. A CDW extracts data from these systems, transforms it into a usable form, and then allows users to view and analyze years of data across a large cross section of patient charts. Although warehouses have existed in healthcare for some time, there are relatively few institutions that maintain patient charts in a CDW. This is, in part, because of the challenges often seen when attempting to warehouse this type of data. These include integrating a diverse set of care practices and a variety of definitions for common data elements like medications, observations, treatments, units of measure, and even unique patient identifiers. In addition, these systems often struggle with a high level of inconsistent and/or incomplete data that must be cleaned up on a regular basis. Unlike other data warehouse systems, CDWs are often expected to gather data around the clock and in a manner that has minimum impact to the performance of the source Clinical Information Systems. Finally, CDWs often have a diverse range of clinical and administrative users. This often leads to a need for a variety of applications and/or tools for viewing and analyzing the data.

CLINICAL information systems (CISs) provide new opportunities to many healthcare providers, including the ability to analyze and to better understand their care practices, costs and effectiveness based on information captured in patient charts. Although this information has always been available to healthcare providers, the cost of mining it from the mountain of paper-based medical records is often prohibitive and prevents a broad analysis of much of the data. As a result, a wealth of clinical knowledge remains undiscovered in these records.

Increasingly, owners of CISs are turning to the clinical data warehouses (CDW) to store and to analyze their data. The CDW allows institutions to make use of the clinical data collected with the CIS. It extracts the data from the CIS, and sometimes other hospital information systems (HIS), transforms it into a usable form and then presents it as information back to the user.

This article provides a technologist's perspective on the challenges of delivering clinical data warehouse solutions. Although these challenges are common for warehouse solutions across all industries, they remain some of the key obstacles facing many healthcare institutions in incorporating warehouse solutions into their information strategy.

WHAT IS A CLINICAL DATA WAREHOUSE

Simply said, a CDW is a place where healthcare providers can gain access to clinical data gathered in the patient care process. This data may include any data related to patient care including specific demographics, vital signs, and I&O (input & output) data recorded for the patient, treatments and procedures performed, supplies used, and costs associated with the patient's care (Fig 1).

Obviously, the warehouse isn't the only place this information may be available. For example, if the patient's chart is maintained electronically, it will exist for some period of time within the CIS. And, once he leaves, his record will also exist either in electronic or paper form in the medical records department. So what makes the CDW unique? First, the role of the CDW is to hold the charted data indefinitely or at least until the data is no longer considered of value to the institution. For this reason, CDWs typically contain large amounts of data, often measured in years. This can mean that the CDW contains thousands, or even millions, of patient records with a wide variety of recorded demographics, diagnoses, treatments, complications, and outcomes.

Second, the data in the CDW are typically reorganized in such a way that it much more efficient to look across patient populations. This is not to say that a researcher could not drill down into a patient's record. However, looking across patient records would be many times faster than it would be querying the CIS, which is organized in such a way to optimize access and updates of a patient record and documents within that chart (Fig 2).

Third, the data in a CDW is typically organized

From Philips Medical Systems, Andover, MA.

Address reprint requests to George W. Gray, BSEE, MSCS, Database Architect, Philips Medical Systems, 300 Minuteman Rd, Andover, MA 01810; E-mail: george.gray@philips.com.

0883-9441/04/1904-0011$30.00/0

doi:10.1016/j.jcrc.2004.08.008

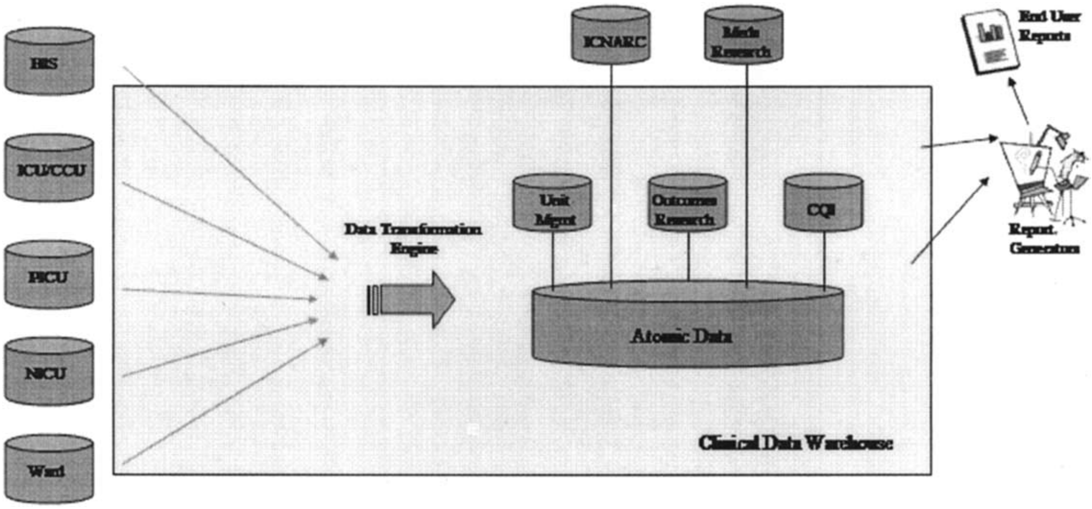

Fig 1. Typical clinical data warehouse topology.

in such a way that is much more intuitive to the novice user than the CIS system. The databases of most CIS systems are organized using a classic entity relation model. In a good entity relation model, redundant data are removed through a process called normalization, allowing for faster updates and less chance of data integrity problems. Entity relation modeling has long been the standard for how operational databases should be designed and is, most likely, the universal way in which all CIS systems are designed. Unfortunately, entity relation models tend to be very complex, with pieces of data distributed across multiple tables.

CDW databases, on the other hand, are designed using dimensional modeling. In dimensional modeling, data are organized in such a way that is much more intuitive and tuned for data access or queries. To do this data is de-normalized, or flattened out into a few significant tables. One byproduct of this is that redundancy begins to appear throughout the model. However, for the average user, the organization of the data is much more intuitive and each query requires the joining of fewer tables. Queries are not only easier to comprehend but also run faster due to the number of tables involved and the organization of data in the dimensional model.

In a dimensional model, details are consolidated into tables called facts. These tables tend to be quite large and grow rapidly as data is added to the warehouse. The fact table in the data warehouse references a number of smaller tables called di-

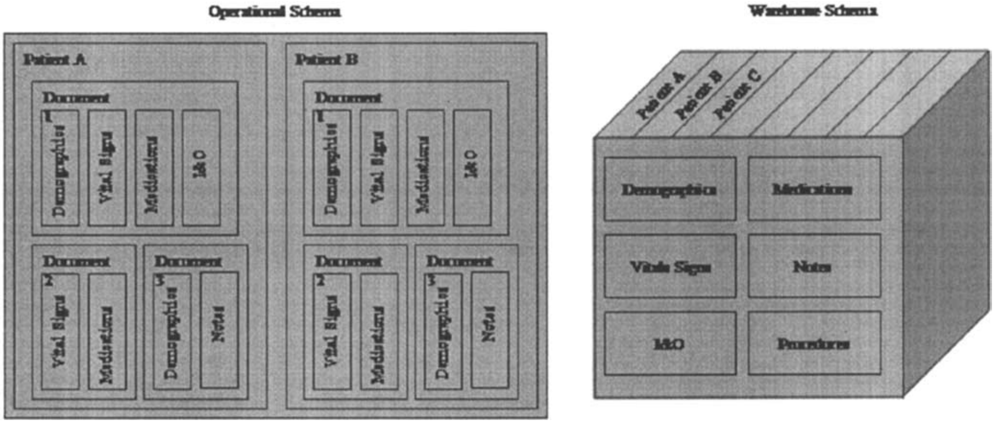

Fig 2. Operational versus warehouse schema structure.

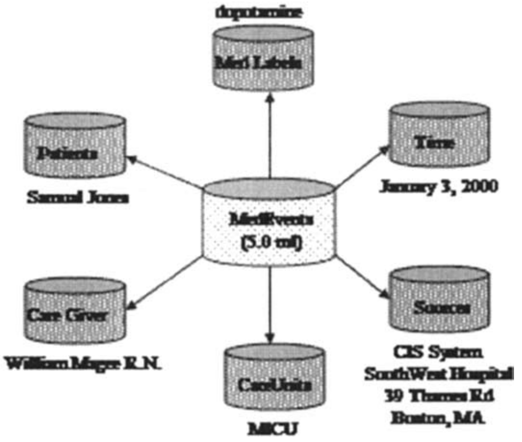

Fig 3. Star schema.

mension tables. The dimensions are typically either slow growing or don't grow at all over time. And relative to the fact table, they are much smaller in size. The dimensions are often connected to the fact in such a way that they model resemble a star, which is why the model is often referred to as a star schema (Fig 3).

In a dimensional model, the facts represent the factual data being stored, where the dimensions represent the key dimensions of the business as well as the questions one might ask about the business. In healthcare, facts include measurements, orders and observations as well as events such as admissions, discharges and transfers. Dimensions, on the other hand, include things like patients, diagnoses, medications, supplies, clinical units, and time.

Finally, the dimensional model is optimized for fast data access. And, as the amount of data increases, the difference in query performance between the dimensional model and entity relation model becomes increasing more significant.

Unlike the CIS, the CDW data are stored offline from the CIS applications. As a result, queries can be run without affecting performance of the clinical applications. Although this is an absolute necessity when allowing ad hoc queries of data, it does have one obvious drawback. The CDW is almost never up to date, often lagging behind the CIS from 1 to 24 hours.

Understanding the Application

Although there are many technical issues that confront the data warehouse architect, none are as critical as the need to start with a clear understanding of how the data and how the warehouse will evolve over time. To do this, the architect typically works with a business analyst, business leaders and expected users of the CDW to understand the key processes of the business and the questions business leaders and other users of the warehouse would ask of those processes. In healthcare, one application area might be unit census, where analysis is conducted on admissions, discharges and transfers by patient demographic, diagnosis, severity of illness, and length of stay. Another application area might be the care planning process, where problems, planned interventions, and expected outcomes are compared against standard care plans and expected results.

Because the warehouse may hold much, if not all of the data collected by the institution, the scope of possible applications for the data is enormous. Therefore, it is important to first assess what questions are asked most often and what data, if made available, would have the greatest impact on the institution's effectiveness and overall business results. Users often hesitate to make these trade offs in fear that some application areas will never be addressed. However, this step is critically important to get a clear understanding of the motivations of the institution and how the success of the CDW will be measured. The CDW will only be seen as successful if it has a positive impact on the performance of the institution and clinical units in which it is installed.

In most industries, this analysis uncovers a set of reports already in use by the organization. Often times, these reports hold the key to what the organization believes is most important. However, the challenge is to determine why these reports are not good enough. The answer is typically because users are unable to drill down into the data and ask follow-up questions about the data in the report. In addition, reports are not always readily available to most users or do not show data that are several weeks old. In healthcare, many see the value of the CDW as a tool that can help present information not currently on these reports such as more detailed outcomes reporting. However, most institutions find it hard to articulate what exactly is of value because most have never had the opportunity to have this information at their finger tips.

This analysis sometimes reveals hidden opportunities for the data and ways the data could be used that were never considered before. This might include providing feedback to people on the front line, helping them provide better service or make better decisions. One example of how warehousing is used in another industry is MCI's Friends and Family (Ashburn, VA), a program that, through a summary of customer utilization trends, was able to target and provide discounts to customers in areas where they were most appreciated. Similar solutions could be provided in healthcare. For example, a clinical unit health monitor could provide directors with a daily view of the "health" of their unit based on certain measurement criteria. Or clinicians could be provided with feedback at the bedside about the effectiveness of certain medications or treatments or comparisons of generic versus non-generic medications based on actual use in the institution.

Once the application areas have been identified, they must be prioritized and a determination must be made as to what applications should be addressed in the first release. Unlike warehouses in other industries, possible CDW applications are quite diverse, forcing its design to be much more complicated.

Because data warehouse solutions are typically unproven in most institutions, it is important for the success and longevity of the effort that the first applications yield the best results in the shortest time frame. In short, the architect and management team must determine what application or applications will make heads turn with the lowest possible up front investment.

Although most technologists give little credence to this challenge, it is by far the most critical decision point in the project. If done well, the project has a much higher chance of success and probability that subsequent applications will be provided on top of the CDW. However, if done poorly, the CDW may either be seen as unsuccessful or be canceled well before it is completed.

Creating a Solid Data Model

Once an application focus has been established, the architect can begin developing a data model that is optimized to support those applications. As stated earlier, the CDW will most likely be designed using dimensional modeling. This will greatly simplify the organization of the data itself and provide much better query performance over time. Because the applications for the data are well understood, the architect can also pay careful attention to tuning the design to best support those questions that will be asked of the data within those application areas.

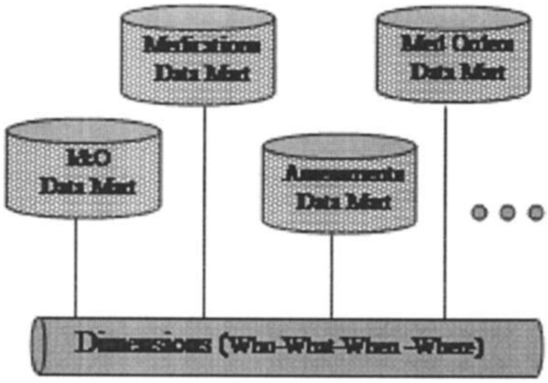

Fig 4. Dimension bus.[1]

The dimensional design will be made up a small number of fact tables, all aligned with a common set of dimensions. Summary or aggregate tables may also be defined that reference the same dimensions. It is important that the architect pay careful attention to the dimensions being defined as well as the attributes associated with each. This is because these dimensions represent a common way in which all questions of the data will be structured now and in the future. This can become much more difficult to accomplish if the warehouse is gathering data from multiple systems because the definition of the data represented in these dimensions might vary from system to system. And, if not careful, the architect may inadvertently change the meaning of some data. For example, a patient's diagnosis on a CIS might have attributes indicating that it is the admitting, discharge, primary or secondary diagnosis. These values as well as the diagnosis itself might change on a regular basis throughout the patient's stay. The HIS system, on the other hand, might define the patient (one of our dimensions) as having an attribute of "diagnosis," which indicates diagnosis on discharge only (Fig 4).

The architect must be careful when defining the dimension's attributes even when the CDW has a single source of information (ie, the patient's age). Knowing that many questions are asked relative to a patient's age or age group, an architect may convert the patient's date of birth to an age upon admission and store this value as an attribute of the patient dimension. This works well when considering adult or pediatric care but becomes a significant problem if the CDW is extended to support neonate patients. Although this may seem like a obvious problem, it is typical of what happens when a technologist with a limited understanding of an application domain begins modeling the CDW. As the model becomes more complex, as in the modeling of Infusions and Drip Medications, the possibility that the data may be misrepresent will increase.

Developing a consistent meaning for all data in the CDW can be particularly challenging. Often times, the way in which something is charted is not always consistent with the way the users of the CDW wish to see it reported. A good example is patient outcome. In the CIS, a patient's outcome might equate to his discharge disposition. However, in the CDW, the outcome might include his discharge disposition as well as a number of measurements and observations recorded on or around the point of discharge. If this consolidation is necessary to support the CDW report, the architect must now take into consideration which distinct measurements should be reported and whether the clinicians charting these values understand their use in the CDW report.

Another data modeling challenge of the CDW is providing ways to cross correlate patient facts at any point in time. It is very common for a user of a CDW to ask questions like "How many ARDS, ventilated, white, males, over the age of 40 received doputamine each month in the MICU?" In this question, adult respiratory distress syndrome, ventilated, white and male all appear to be properties of the patient. However, they are not. Instead they are facts associated with the patient at a particular point in time. To complicate matters worse, CISs typically don't require users to chart when a patient is being ventilated or other facts like their date of birth. Sometimes this requires that the fact be derived from other facts or simply results in a misrepresentation of what is actually occurring in the patient population. In short, the architect must be sure that data are never misrepresented to the user.

As with any data warehouse solution, another challenge in designing the data model is determining the granularity of the data. At one end of the spectrum, users may want to see the data in its most atomic form. For example, a user may want to see each medication administration, when it was administered, how much was given, who gave it and who it was administered to. This, of course, is how clinicians are accustomed to seeing this data when caring for the patient but may not be what they are interested in when viewing this data retrospectively across an entire patient population. On the other end of the spectrum, the data can be summarized, showing the total amount administered by medication, treatment, and patient. This is the more typical representation of what would appear on a report but can limit a clinician's ability to drill down into the data. Again, the questions must be asked, "What are the near term and long term applications for this data?" Because of the disparity of possible applications for the data in the CDW, most store the data both its atomic form as well as in the aggregate.

Given that millions of rows can exist in the CDW fact tables, it is advantageous to define aggregations to support the targeted applications through summarized views of the data. However, to aggregate the data, the data must be additive across one or more dimensions. Unfortunately, in the CDW, many of the measurements and observations recorded about a patient are not additive across any dimension, making this data near impossible to aggregate. This is particularly true with vitals signs, which represent the largest volume of patient data. Imagine showing the total number of heart rate measurements or the sum of all blood pressures. This data has no meaning, except in its atomic form or compared relative to one another.

Transforming the Data

At the same time the model is being defined, the architect must also design the mechanism used to move data from its source systems to the CDW. This is typically referred to as the ETL (Extraction-Transformation-Load) service or simply the transformation service. A transformation service can be both expensive to develop and to support. For this reason, many organizations purchase programmable transformation engines that facilitate the transformation process and allow engineers to both configure simple transformations as well as develop custom transformation functions. The primary drawback of these engines is their price, typically costing $100,000 or more.

The issue of technology aside, the primary challenges of designing a good transformation process include:

1. Reducing the impact on the source operational (ie, CIS) systems; and
2. Minimizing the time required to transform and store the data in the CDW.

When the transformation engine runs, it queries each source system for large amounts of data. If not careful, a poor transformation engine design can have a significant impact on the source systems performance. Across the industry, most data warehouse transformations occur once a day in the evening. This is typically the time when little or no activity is occurring on the source system. For the CDW, however, there is never a time when the CIS or HIS is not in use. As a result much more care must be taken to ensure that it impact on the source system is minimized.

During the transformation process, data are extracted from the source system, transformed to fit within the CDW dimensional model and then loaded into that database. In addition, data are often summarized and rolled up into aggregation tables at this time as well. In many industries, the time required to perform this transformation is not critical because the transformation typically occurs in the evening and data is not viewed until morning. However, in some institutions, it is a requirement to keep the CDW up to date on an hourly basis. As result, the transformation process has a maximum budget of 1 hour to complete. However, because the transformation of data is typically very CPU intensive, it will impact query performance of the CDW itself. As a result, the real transformation budget is probably less than 10 minutes if run every hour. For the architect, the need for timely information must be balanced against the need for complex transformations and subsequent aggregations of the data.

For CDWs that need to integrate information from multiple source systems, the architect must also consider the need to synchronize this data before storing it in the CDW. For example, consider the case where ADT information is being retrieved from an HIS system and integrated with orders from an order entry system and measurements, observations, notes and care plans charted on the CIS system. Before processing the CIS data, the transformation engine would need to process the HIS and then the order entry data. If the ADT or order entry data were unavailable for some reason, the transformation engine would need to decide whether to proceed or wait until all data is available from all source systems. This would make the transformation process only as reliable as the combined reliability of the three systems. On the other hand, if the transformation proceeds, it might present the data in an inconsistent or erroneous state until the required data arrives.

One of the most insidious challenges of many data warehousing project is dirty data. This is because fixing it is often outside the direct control of the architect. Often times, dirty data are the direct result of bad data being entered in the CIS or HIS system or required data not being entered at all. The source system typically does not enforce all the constraints required of the data in the CDW applications. The reason for this is that the CDW is supporting a different application for the data and therefore tries to impose new constraints on the data. As a result, data might be missing, inconsistent or even wrong based on an agreed upon definition of the data.

Dirty data can also arise from inconsistent use of labels in different source systems. For example, the architect may need to determine whether HR is the logical equivalent of Heart Rate and then determine which one to represent in the CDW. Gender might vary from M and F to Male and Female across the source systems or even within a specific CIS system as configurations change over time. The architect must determine whether these differences should be resolved during the transformation process.

Dirty data can also be caused when data are transformed in unnatural ways and no restrictions are placed on the source system to enforce a certain charting practice. For example, let's assume the CDW keeps track of when a patient is intubated and extubated. However, because the source system does not require that users chart when an intubation or exbuation occurs, the transformation process derives this information by looking at charted parameters such as O2 Delivery Mode and Mode of Extubation. Although these might be reasonable assumptions, they are also the potential source of more dirty data unless it is enforced that all intubated patients (even those intubated prior to admission into the unit) have these 2 parameters charted and at the times representative of when they were intubated and extubated.

Presenting Information

At some point, data must be presented back to the end user as information. A variety of tools are available to users to achieve this. Each of these tools satisfies a different type of user and budget. Because for many, the user interface of the CDW is actually the one presented to them by these tools, selection of the right tool is very important. The best tool varies depending on the expertise of the end users and the applications for the data. In many cases a variety of tools are required. These include report generators for both dynamic and static reporting, OLAP/Data Analysis tools, data browsers, data mining tools and, of course, custom applications.

REFERENCE

1. Kimball R, Reeves L, Ross M, et al: The Data Warehouse Lifecycle Toolkit. New York, NY, John Wiley and Sons, 1998