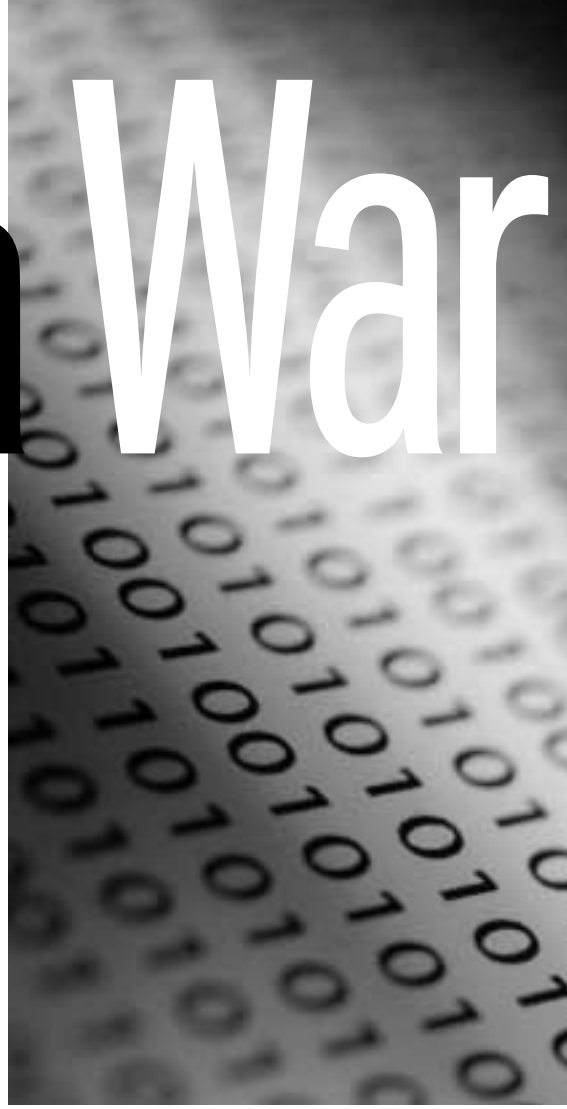
The Data War
Keeping It Simple



MIT shares valuable lessons learned from a successful data warehouse implementation

by **Scott Thorne**

At MIT, we have implemented a data warehouse to provide a reporting and data distribution environment for departments, labs, and centers as well as to support some of the reporting needs of central departments. In addition, the warehouse acts as a hub, facilitating the exchange of information between the institute's administrative information systems. Taking these functions together, the warehouse serves as MIT's enterprise information infrastructure.

Our data warehouse has been in existence for about four years, but only in the past year, as we added information from our SAP financial system, has it reached maturity. Currently the warehouse includes personnel, financial, purchasing, telephone, student, and award information. More than 1,200 registered users participate in about 3,500 sessions a month.

This article explains why our data warehouse is important to the MIT community, describes its basic functions and technical design points, and shares some of the non-technical aspects of our data warehouse implementation that proved to be important. Our experiences in building and maintaining MIT's data warehouse and the lessons we have learned in the process may be helpful to you as you plan your campus data warehouse.

Why a Data Warehouse?

The basic vision for our data warehouse is to make information at MIT more accessible and easier to use for a diverse group of users in the institute community. If people can access data easily, they will spend less time gathering information and more time analyzing it. Having information from several different sources allows people to create reports easily and use information in new and creative ways. In the past, similar reports would have taken days to construct, and even if individuals invested the time and energy to put together information for themselves, they could not have easily shared it. With the technologies available today, once a user figures out a good way to look at the information (by generating a report definition in the query facility, for example), they can share it easily with others.

By having information available in this way, users are able to combine warehouse information with local information. This is a powerful concept because it frees users from having to recreate information available in the warehouse so that they can focus on information unique to them. In this way the warehouse and local systems complement each other.

Making data more accessible also serves to improve data quality over time. As people use the data, errors can be corrected as they are found. Additionally, using the information for

more purposes helps improve the design of future systems.

Warehouse Functions and Characteristics

Our goal is to make all appropriate administrative information available from the warehouse. The two basic requirements we have for determining what information goes into the warehouse are that the information is of use to more than one group of people and that it has a source system of record.

MIT's data warehouse has three key characteristics:

- *Many sources*. The warehouse integrates data from various administrative systems and stores them in one location.
- *Read only*. The information represented in the warehouse is maintained in other systems, called "systems of record."
- *Restructured information.* The warehouse presents data in a simple form so that reports are easy to construct and the warehouse easy to use.

Since the community has diverse needs, the warehouse has been designed to support three separate access mechanisms:

1. *End-user query and reporting tools*. MIT has obtained a site license for BrioQuery, one of a large number of commercial products in this area. Ad hoc reports are very easily created. Standard reports can be defined and shared easily through the Web or an e-mail attachment.

2. *Creation of file extracts*. Data files can be created and transferred from the warehouse to local systems. In addition database links, snapshots, and other database-to-database transfers can be done.

3. *Custom programs*. Programs can be written to access the warehouse using structured query language (SQL). These can range from simple perl scripts which just extract data to full applications written in Powerbuilder, C, Java, and so forth.

EASY TO USE

The success of the warehouse depends on our ability to present the data in as simple a form as possible and to make interactions with the data warehouse as straightforward as possible. To generate common reports, end users have access to data that are in an easy-to-understand and easy-to-use structure. Unlike a traditional transactional system, which minimizes the storage locations of data to make updates more efficient (normalizing), the data warehouse duplicates data where appropriate so that reports can be generated more quickly and easily. Although this strategy uses more disk space, it makes reporting access much easier and faster.

USER ENGAGEMENT

We have a diverse set of users at MIT. Some people need to be able to just push a button to get the standard report they need. Others need to analyze information in a much more dynamic way, asking a series of questions (ad hoc queries) to investigate something. We do not expect all users to learn how to build reports themselves, but we hope they will become comfortable with running prebuilt queries. Then as they use and understand the information they access, they will begin creating their own reports, hopefully sharing this work with others.

A warehouse should ultimately change the way people use information. It takes time for this change to happen, and people change at different rates, so a warehouse needs to evolve over time. At the outset people want to be able to look at the information in much the same way as they are used to seeing it. Allowing them to do this makes them become familiar with the warehouse and have confidence that it is providing the same information

they are used to getting. Later they can begin at their own pace looking at information in new ways, such as in new layouts, at different levels of aggregation, or in exception reports.

QUALITY OF INFORMATION

The implementation of a subject in the warehouse goes through a progression of stages. These stages can be years long. The quality and usefulness of data improve through each successive stage of implementation, from getting accurate detailed data within a subject area, to integrating accurately detailed information among subject areas (for example, combining personnel and payroll information), to creating useful summary, aggregate, and history information.

The quality of the information in the warehouse must be high for users to perform their reporting needs. There may be problems with data quality because some of the data being delivered have not been accessible to the community before and have not been previously reviewed and corrected. Getting the data published and having a well-documented procedure for making corrections to data should go a long way towards making the warehouse information accurate.

Because the warehouse is read-only and not updated during the day, consistent reports can be generated from a stable data set. Users generating reports can be assured that they are obtaining information from stable data.

Technical Design Points

For easier reporting, our data warehouse reconstructs information that comes nightly from source systems such as SAP. The warehouse has row-level access control that is driven from an enterprise authorization system called Roles. Other design features are a metadata-driven data conversion and load process, distributed report creation, encrypted transport, and a system of automated integrity checks and controls.

STAR SCHEMAS

We have organized the information in the warehouse in star schemas. Star schemas consist of a central fact table joined with several dimension tables. The fact table is extremely large and has all the transactional data and numbers; the dimension tables are smaller and have descriptive information. An example of a fact table is our financial_detail table. Examples of dimension tables are time_month, gl_account, and cost_collector. The dimensions allow a user to sort, group, and limit the fact records easily. Since the dimension tables are smaller, users can display all possible values of a field when deciding how to limit their request. Users use many of the same dimension tables with different fact tables, so although it initially can seem overwhelming most users can become familiar with one set of tables quickly. This type of schema is both flexible and efficient.

Star schemas have other benefits. Since all the amounts and details are in the fact table, this is the only one that usually needs row-level access control. The dimensions can be left open and viewable to anyone, since they have only descriptive information like cost collector names and numbers.

Another feature made possible by star schemas is a flexible way of viewing historical information. For example, when viewing old financial information, do you want to see the numbers with the account information the way it was when it happened or recast in the way you are organized now? Since the descriptive information is in the dimension table it is possible to create snapshot versions of dimension records periodically. Then by joining the dimension table to the fact table using different key fields, you can get either view from the same fact table.

INTEGRITY CHECKS

Knowing the information in the warehouse is a truly accurate reflection of the source system is extremely important, as people are going to rely on the warehouse for reporting. Users need accurate clear definitions of all the data presented in the warehouse. Beyond the definitions, they also need easy access to information concerning, for example, when the data were last loaded and from what source, how to report errors, or how to get changes made to a particular field and record.

OPENNESS AND FLEXIBILITY

Direct SQL access for users gives the data warehouse openness (ability to use a variety of tools) and flexibility (putting information together in new ways). Many warehouses are designed with a front end (such as the Web) and no direct SQL access. Users will ultimately be limited in how they use such a warehouse. Viewing information is fine, but many users actually need to capture the data to manipulate them further on their own or combine them with local information. Using SQL does not preclude us from presenting the information via the Web or some other front-end application in the future.

METADATA

Metadata or data about data is a critical component in a warehouse environment. Providing metadata has many benefits. Users need to know what information is available, how it is organized, and what it means. Data transformation and loading tools or programs need to know where the data come from and how they need to change. Both users and conversion programs or tools need to know the structure the data are going into.

Since we built our own tools for this project, we also designed our own structure for maintaining metadata. This allows us to do all the maintenance in one place and avoid problems of different representations of metadata that are out of synch. It also allows us to design generic programs for the conversion and loading of the warehouse that get the specifics from the metadata. This enables a lot of code reuse and minimal code changes as we make changes to conversions and loads.

Most of the time we actually do not write new code to implement a new subject; it is simply a matter of creating the metadata correctly in order to automate the loads with our existing generic programs. If an error occurs most likely it is caused by either a data or metadata problem. Having metadata makes it easy to change the way information is being loaded without having to recode. The same software gets reused in many different places making it unlikely that a problem will remain undetected in this area.

ACCESS CONTROL

Institute data must be handled with the proper security and access control. Data administration policies need to be worked out in advance and simplified as much as possible. Our warehouse design maintains security at the database level. All transmissions of data across the network are encrypted. Additionally, for users to view only the information they are allowed to see, such as their department's information, the warehouse presents most data through "views." With this scheme, users see what looks like a table, for example, "employees." However, in actuality, this is a view that shows each user a different set of data depending on the access control they have been granted.

This is accomplished by creating access control tables, which are joined with the base data tables in the view. The access control table has the user's username as one field and some identifier in the other. The identifier is also present in the base data table. The view only selects rows from the base table where the identifier matches the identifier in the access control table and the username matches the person connected. This way a single view definition could yield all Department B employees for Sue, while giving Tom only Department C employees.

Not only is the maintenance reduced but this approach allows the users to share reporting templates and queries easily, since the objects they are using are the same. Because the access control table structure and content are so simple, these tables can be created easily or imported from another source. Using a view, which joins two tables, will impact performance but this can work well on fact tables with millions of rows and access tables with hundreds of thousands of entries as long as the information is structured correctly.

ROLES SYSTEM

The variety of different systems that sometimes contain similar information make implementing rational, consistent access control difficult. When each system has separate mechanisms for establishing access it takes a great deal of work to keep them in synch. In order to simplify this area at MIT, we designed a central authorization system called Roles. This system allows for the maintenance of authorizations in a single system that can then drive the authorizations in both the system of record and the warehouse. In our case the financial reporting authorizations are maintained in Roles and fed to both SAP and the warehouse, which always keeps them in synch. Furthermore, the interface for maintaining them can be distributed out to the departments, so that the people who know what the authorizations should be are the ones maintaining them.

An enterprise access control infrastructure has the advantage of:

- Achieving consistent access control over multiple systems
- Distributing the maintenance of access controls to a person close enough to know and care whether the authorizations are correct
- Having a simple way to maintain and view authorizations across many systems
- Granting authorizations at the highest possible level but enforcing them at a detail level, which minimizes the changes to authorizations due to account or organizational restructuring.

Training Is Critical

Proper training is of the utmost importance to the success of a warehouse project. At MIT we train users not only on how to use the software tool but also about what the information means and how to use it and when it is appropriate

to use the warehouse rather than to query the transactional system.

Some users are more self-sufficient than others, so developing different ways to train is important. Training methods we offer include standard hands-on classroom training, custom-developed training for a group of users with common needs, open lab sessions, and one-on-one help. Overall we have used probably half our resources over the last two years to provide training and we expect this to continue for the foreseeable future.

We offer formal hands-on classroom training to learn Brio, our warehouse software tool. We encourage the majority of users to attend two such classes (beginning and advanced) that are offered regularly by our IT training staff. We developed custom training for some special groups such as internal auditors and personnel from the budget office so that they could learn with information and examples that they are familiar with. There are also quick one-hour sessions for users who just want to run prebuilt reports. Twice a week we have a couple of hours reserved for an open lab session, where users can come in with reports they are working on and get help from the warehouse staff. It is very helpful to us, as well, to see what people are trying to do as we get many ideas of fields to add and other improvements from this interaction.

What information you base your training on is very important. If the users are looking at real information with which they are already familiar, then they learn more quickly. Some people report on data that are too sensitive to be used as a basis for a regular training exercise; in such cases, we produce test information that can be used in a classroom setting.

Some Pitfalls to Avoid

From our experiences, there are a few areas that are especially important to the success of a warehouse project.

Not restructuring data is one of the most common mistakes made in warehouse implementations. If you just move the data to the warehouse without changing the structure all you really achieve is offloading some reporting from the transactional system. Some people argue that if you do not keep the data the same, how will you know that you have an accurate representation of the source system? There are other ways to ensure integrity, and the benefit of doing this restructuring work initially will save enormous amounts of report development time and execution time, as well as allow mere mortals to construct queries.

Managing expectations is a pitfall in every IT development project, but the variety of subjects, users, and uses makes expectations especially hard to manage in a data warehouse project. The traditional advice to "plan for success" has real meaning in developing a data warehouse. While it is important to demonstrate value with some early results to get user buy-in quickly, once you have demonstrated that value, users will immediately want access to all kinds of information. If you have not worked out scaleable processes for handling the new data and increased numbers of users, you will not be able to meet those expectations. Unfortunately this means taking more time up front to design processes that will scale. If you decide to take a prototype approach for a quick win, you will need to manage user expectations as to how soon they can expect a fully operational warehouse with all the information they need.

A lot of time and attention needs to be given to all the various administrative systems stakeholders, who must fully participate and find the warehouse beneficial for it to be successful. Central functional departments traditionally have the role of providing access to data. Getting them to buy in to a warehouse without compromising the warehouse design principles needs to be done on a department-by-department basis. This part of the process cannot be rushed because without the cooperation of these stakeholders the warehouse effort is doomed to failure.

After Our Initial Success

The data warehouse at MIT is now enjoying some success. The usage has been climbing steadily this past year as more and more people have incorporated use of the warehouse into their normal work. The central offices that are the information providers are finding many warehouse advantages. They do not have to write and maintain as many feeds now, since people can get information directly from the warehouse. Also, having other ways to analyze their own data helps uncover problems sooner. Some of their common reports can be run against the warehouse in far less time because of the more efficient reporting structures.

There are many benefits to a warehouse, the most obvious being easy, flexible, and integrated reporting. The greatest benefits, however, can be by-products of the effort—improving information quality, rethinking information needs, articulating information access policies, and building consensus around the meaning of certain terms and information.

A major challenge to information quality results from combining information from several different source systems as we have done with our data warehouse. In the past there was rarely a need for information from these systems to be prepared to combine with data from another system. Thus we have encountered such problems as unique identifiers on records and similar but different data elements. So while the majority of key subject areas exist in some form in the MIT warehouse, most of these subject areas still cannot be easily joined to others because of these problems. We plan now to turn our attention to this important aspect of data administration. *e*

Scott Thorne (thorne@mit.edu) is the data administrator at MIT.

Recognizing the Winners

2000 EDUCAUSE Award Programs

EDUCAUSE
Transforming Education Through Information Technologies

Leadership Awards

program sponsored by SCT

Excellence in Leadership

- for extraordinary effectiveness, influence, statesmanship, and lifetime achievement, on both individual campuses and the wider higher education community

Ira H. Fuchs
Vice President for Research in Information Technology
The Andrew W. Mellon Foundation

Leadership in the Profession

- for exceptionally effective leadership in campus information technology use and management, and mentoring of other professionals

David L. Smallen
Director, Information Technology Services
Hamilton College

Jerry Niebaum
Assistant Vice Chancellor for Information Services
University of Kansas

Leadership in Information Technology

- for visionary achievements and effectiveness in identifying and advancing technology directions for the various needs of higher education

Jeffrey I. Shiller
Network Manager
MIT

Excellence in Campus Networking

program sponsored by Novell, Inc.

Winner

Wake Forest University

An exemplary blend of standardization, decentralized support, and allowance for customized computing needs that creates an environment of low-cost ownership and exceptional acceptance of technology

Honorable Mention

University of Northern Iowa

EDUCAUSE Quarterly Contribution of the Year

program sponsored by SCT

"The Catalyst Project: Supporting Faculty Uses of the Web . . . with the Web"

A fresh approach to support of instructional technology based on a very collaborative process

Mark C. Donovan
Strategic Marketing Manager
RealNetworks
and Senior Research Fellow, Center for American Politics and Public Policy
University of Washington

Scott Macklin
Director, Program for Educational Transformation through Technology (PETTT)
University of Washington

Systemic Progress in Teaching & Learning

program sponsored by Eduprise

Winner

University of Washington

Vibrant, cross-campus programs and collaborations that encompass experimentation, implementation, assessment, and distribution of mainstream, open-standards-based tools to improve teaching and learning through technology

Honorable Mention

Seton Hall University

Honorable Mention

University of Technology, Sydney

Exemplary Practices in Information Technology Solutions

program sponsored by PeopleSoft, Inc.

Winner

HoyasOnline
Georgetown University

The transformation of a standard alumni directory into a unique Web-based resource to broaden communications and foster community among the university's 125,000 alumni

Winner

PAWS—Personal Access Web Service
Louisiana State University

A one-stop shop to university services that delivers 55,000 unique intranet portals, each customized to dynamically reflect the individuals' current relationship to the university

For information about these achievements—and about award programs for 2001—www.educause.edu/awards/awards.html or e-mail us at awards@educause.edu

The EDUCAUSE program of awards brings peer endorsement and visibility to achievements in many arenas. Judged and ratified by professionals with expertise in the fields they are evaluating, each award is an affirmation of merit.