Data Warehouse at MIT: Strategy Document

On this page:

Vision
Definition and Functions
User Engagement
Design Points
Quality of Information
Easy to Use - Simplicity
Flexible - "Open"
Reliable
Security
Low Operating Cost
Maintainable
Evolutionary
Scalable

Vision

The basic vision of the Data Warehouse is to make information at MIT more accessible and easier to use. If people can get 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. Once similar reports would have taken days to construct. This allows time for information to be used in new and creative ways. In the past, even if individuals invested the time and energy to put together information for themselves, it was not easily shared. With the technologies available today, once a user figures out a good way to look at the information (by generating a report definition in BrioQuery, for example), they can easily share it with others. Also, by making information available in this way, users should be able to combine Data Warehouse information with local information. This is a powerful concept, because it frees the users to keep track of the information unique to them, and not recreate information available in the Warehouse. In this way the Warehouse and local systems compliment each other. Making data more accessible also serves several other goals. First, it will 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 will help us improve the design of our systems in the future, so that we will have the information that we need on hand.

Definition and Functions

The Data Warehouse is a database with information coming from many areas, which is structured in a way to make it easy to use the information.

- Many Sources; integrates data from various administrative systems and stores them in one location.
- Read Only; is a read only database. Information represented in the Warehouse is maintained in other systems, called "systems of record".
- Restructured Information; presents data in a simple form so that reports are easy to construct and the Data Warehouse is easy to use.

The main purpose of the Data Warehouse is a reporting and data distribution environment for Departments, Labs and Centers at MIT. In addition to this, the Data Warehouse will support some of the reporting needs of Central departments. Also, the Data Warehouse acts as a hub, to facilitate the exchange of information between systems. Taking all of these functions together, the Data Warehouse serves as the enterprise information infrastructure.

The Data Warehouse is not the place to solve problems like "Did this invoice get paid?" This type of
question should still be directed to the transactional systems such as SAP. A warehouse system is not meant to take the place of the transactional system, but to complement it. It will take some training for users to understand when they should be using the Data Warehouse and when it would be better to query the transactional system.

The goal is to put all administrative information in the Data Warehouse. The two basic requirements we have for determining which information goes into the Data Warehouse are:

- That the information is of use to more than one group of people.
- That the information has a source system of record.

Since the community has diverse needs, the Warehouse is being designed to support three separate access mechanisms:

- 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 email attachment.
- Creation of data extracts: Data can be created and transferred from the Data Warehouse to local systems via database links, snapshots, FTP, etc.
- Custom programs: Programs can be written to access the Warehouse using SQL. These can range from simple Perl scripts which just extract data, to full applications written in Powerbuilder, C, Java, etc.

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 don't expect all users to learn how to build reports themselves. What we hope to do is get users comfortable with running pre-built queries. Then when the need arises, they can learn to modify the reports to suit themselves. As users use and understand information they will start creating reports themselves, hopefully sharing this work with others.

Design Points

The design of the Data Warehouse has many aspects. As in most designs the considerations need to be balanced. For example; simplicity vs. flexibility and functionality; every time we allow another way to do things we make the system a bit more complicated.

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.

The stages are:

1. Getting accurate detailed data within a subject area.
2. Integrating accurately detailed information among subject areas, e.g., combining Personnel and Payroll information.
3. 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 has not been accessible to the community before, and it has not been reviewed and corrected previously. 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.

Another problem with data is that we are attempting to combine information from several different sources. These source systems rarely had a need in the past to make sure their information was prepared to be combined with data from another system. Therefore, problems such as having unique identifiers on records, or having several similar but different data elements will cause some problems. The solution is rethinking and altering of some of the source systems along with the Warehouse, which will take some time to do correctly.

Knowing the information in the Warehouse is a truly accurate reflection of the source system is extremely important, is people are going to rely on the Warehouse for reporting. The Warehouse implements several methods to try to assure this. ...

Users need accurate clear definitions of all the data presented in the Warehouse if they are to make use of it. Beyond the definitions, they also need easy access to information concerning, for example, when the data was last loaded, where it came from, how to report errors, or how to get changes made to a particular field and record.

**Easy to Use - Simplicity**

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 simple and straightforward as possible. To generate common reports, the end users will 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 more easily. Although this strategy uses more disk space, it makes reporting access much easier and faster.

**Flexible - "Open"**

The Warehouse will need to serve a diverse user community. Users will use information in different ways. The design of the Warehouse needs to allow for this. Unfortunately, creating a system with as much flexibility as possible usually means that simplicity gets compromised. We're striving to achieve the proper balance between flexibility and ease of use.

Direct SQL access for users gives the Data Warehouse the 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 the Warehouse. Viewing information is fine, but many users actually need the data to manipulate further on their own or combine with local information. Using SQL doesn't preclude us from presenting the information via the web or some other front-end application in the future.

**Reliable**

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. The service level should be well understood. For example, there will be times when the Warehouse has a service outage due to technical problems. Users will be notified and told when they can expect to have the Warehouse back in service.

**Security**

Institute data must be handled with the proper security and access control. The Warehouse design, therefore, maintains security at the database level. All transmissions of data across the network is encrypted. Additionally, for users to view only the information they are allowed to see, such as their department's information, the Warehouse will present most data through "views." With this scheme, users
will see what looks like a table, e.g., "employees." However, in actuality, this is a view that shows each user a different set of data depending on the access control that has been granted.

**Low Operating Cost**

The Warehouse is designed to have a low cost to run. Everything which can be automated will be. This frees the Warehouse team to continue to add to and improve the Warehouse.

**Maintainable**

The Warehouse is designed to change and be altered over time. The Warehouse loads and data transformations are driven from data in the database itself. The "metadata" makes it easy to change the way data is being loaded without having to recode. The same software gets reused in many different places making it unlikely that a problem would remain undetected in this area.

**Evolutionary**

We hope to always be adding to the information to the Warehouse, but not taking away or changing significantly what it there.

**Scalable**

We assume that the Warehouse will contain more information and be used by more and more people. We try to make sure that our design will hold up as the volume of information and usage grows.