CRITIKAL

ESPRIT Project Number 22700

Attar Software

Gehe

Lloyds TSB Group

Parallel Applications Centre

University of Stuttgart

D2.4 Data Warehouse Design: The Impact of Data Mining

PAC/CRITIKAL/D2.4 Version 1

Paul Allen
Parallel Applications Centre

18 January 1999

Executive Summary

This reports assesses the impact of data mining on the data warehouse and has been compiled from work performed in the part EC-funded CRITIKAL project.

The report tackles the questions "What considerations need to be made when the data warehouse is designed?" and "What changes should be made to the data warehouse to prepare for data mining?". The report is aimed at senior data warehouse and IT strategy personnel.

The level of detail to which data is stored in the warehouse has a critical effect on the degree of data mining that can be performed. It is recommended that atomic data is stored on the data warehouse, rather than the use of pre-aggregated data. This provides the necessary flexibility which is required for data mining analysis.

Warehouses contain dynamic data, typically transactional information, and 'static' data such as reference data. This reference data is not truly static, however — treating the data as static can lead to errors when data mining. It is recommended that some form of versioning system is implemented in the warehouse for the reference data. This versioning system enables a point in the past to be recreated. The versioning system also allows "like for like" comparisons to be made between present and historical data.

It is recommended that use is made of warehouse metadata in order to locate the optimum data for the data mining exercise. Metadata should be recorded in the warehouse to track when an interpretation of a column changes.

It is recommended that metadata is used to manage intermediate results that come from a data mining exercise. These intermediate results and the accompanying metadata can be stored on the data warehouse to aid future analysis.

It is recommended that derived and deduced data, either produced as a result of data mining or produced for data mining, are stored on the data warehouse. The derived and deduced data should be stored in separate tables to the base data, and joined when required using a join key. This approach will facilitate maintenance and ensure that the derived data does not obscure the base data. Metadata should be used to document any transformations used and assumptions made in deriving or deducing data.

For data mining, flexibility is important and for this reason it is recommended that data is stored in normalised form in the data warehouse. Star or snowflake schemas can be used, but as an additional layer on top of the warehouse, or in a dependent data mart.

It is recommended that processing resource is provided to run the data mining control and algorithms. High performance can be obtained through either:

- using a high-performance application server;
- using computing resource of the database machine as a high-performance application server; or
- using the database directly to obtain performance.

Indexing is often not useful for data mining due to the unpredictable nature of the data mining process and the frequent need to aggregate over the entire data set. Data partitioning can be useful for efficient extraction of large contiguous sections of data, particularly date/time partitioned transactional data.

Contents

1 Introduction

This reports assesses the impact of data mining on the data warehouse and has been compiled from work performed in the part EC-funded CRITIKAL project.

The report tackles the questions "What considerations need to be made when the data warehouse is designed?" and "What changes should be made to the data warehouse to prepare for data mining?". The report is aimed at senior data warehouse and IT strategy personnel.

The subject of data mining has received a lot of press and many organisations with a data warehouse have plans to exploit the power of data mining. However, the core applications and quick wins from warehousing are obtained from simpler analyses; data mining often represents a long-term objective for data warehousing projects. As a result, it is rare for data mining to be planned for in the early stages of a data warehouse and mining projects tend to be implemented on warehouses that do not have the optimum infrastructure in place for data mining projects..

The objectives of this report are to provide guidelines to make data mining as efficient as possible (in terms of the process, human resource and computing resource) and as useful as possible (in terms of positive impact on the business).

There is a continuous spectrum of data analysis approaches, from simple reporting through analysis to data mining. Many issues highlighted in this report also apply to many analysis methods and (to a lesser extent) all data warehouse analyses. This report focuses on issues that differentiate data mining from other forms of analysis.

The report does not claim that all the guidelines need to be followed before it is possible to do any data mining — many organisations are successfully mining with no warehouse at all. However, the guidelines will allow the maximum benefit to be gained from the data mining process.

2 Scope

The scope of this report covers the following areas:

- the data that should be stored in the warehouse;
- data warehouse management, including cleaning, metadata and derived data;
- the logical storage of data; and
- the physical storage of data.

This report is not a general purpose data warehouse design guide, and many principles of good data warehouse design also apply to data mining. These principles for warehouse design are widely available (for example, "Building the Data Warehouse" by William H Inmon [1]).

3 Characteristics of data mining

This section describes the characteristics of the data mining process and the generic requirements that data mining places on the data warehouse.

Data mining is a human driven iterative process [2][3]. This has consequences for the data warehouse — the warehouse must be able to support train-of-thought analysis, hypothesis testing etc. This leads to a requirement for the warehouse to be flexible, data-rich and performant.

Typically the final data mining algorithm is performed on a single "data mining" table. This table is produced using transformations and aggregations on the base data. Often derived fields are produced. This transformation is a key part of the data mining process [4]. Often it is a manual process and the physically elapsed time for locating, migrating, transforming data is orders of magnitude greater than the computing time involved. It is important that effective tools are used to support this process.

Often a data mining exercise will involve the analysis of a set of data collected over a fixed time period. Data pertaining to the business functions relevant to the problem is gathered together and analysed during the data mining process. For example a customer segmentation exercise may involve the study of transactional, customer and product data relating to the first quarter of 1998. It may be necessary to look at environmental data, such as competitor data — in this case the relevant competitor data from the first quarter of 1998 is required. In other words, the analyst needs to recreate the conditions, both internal and external to the business, of the time period in question. This has repercussions for the data warehouse.

In terms of the query impact on the database server, the data mining profile is similar to the analyst power user. However, as the queries are automatically generated, the main differentiating factor is the higher volume of queries. Additionally, the queries may be more complex — it should be noted that analyst power users can generate extremely complex queries, particularly with the aid of analysis tools generating automatic queries.

When useful results are discovered through data mining, it is essential that they are exploited by some sort of business action in order for the organisation to gain advantage from the mining exercise. It is important to measure the impact of this action on the business in order to assess the value and return on investment of the data mining. It is important that the necessary processes are put in place early on to measure and monitor key attributes of the business using the warehouse, in order to fully assess the impact of data mining.

4 Data decisions

This section covers the nature of the data that should be stored on the warehouse.

4.1 Level of detail

The level of detail to which data is stored in the warehouse has a critical effect on the degree of data mining that can be performed. It is recommended that atomic data is stored on the data warehouse, rather than the use of pre-aggregated data. This provides the necessary flexibility which is required for data mining analysis.

As described in Section 3, the data mining process often involves aggregation to produce the data mining table. In some cases it is possible to create a first pass data mining table using pre-aggregated data. However, data mining is an iterative process and during the course of the analysis it is likely that new hypotheses will be generated. These new hypotheses will require the user to re-aggregate the base data in order to add fields to

the data mining table. If the base data is not available on the warehouse, it is not possible to perform this new analysis.

As an example, consider a telecommunications organisation performing a customer segmentation exercise. The analyst may have a hypothesis that the parents of school children using the Internet represent a valuable customer segment, which the analyst may wish to verify be examining the after-school use of the telephone during term time. This will require an aggregation of the customer base to produce the field "average time using telephone after 4pm during term time". This is not possible using pre-aggregated data, unless the organisation had already identified this parameter as a valuable metric. As one of the goals of data mining is to identify and verify new hypothesis, it is important that the atomic data is stored in the warehouse.

Basket level sales data represents the atomic level of information for data mining in the retail sector; individual transaction level data for the financial sector, and call description data for the telecommunications sector

In scientific and engineering sectors, the equivalent to transactional data is sampled continuous data. This may take the form of gas flow rates, temperatures, pressures etc., measured at regular intervals. Here there is no concept of 'atomic' as the sample intervals can be arbitrarily small. It is not possible to give precise guidelines for data mining in this instance, as the appropriate sampling rate will vary, on a case-by-case basis depending on the dynamic nature of the system.

Ideally (for data mining purposes) the atomic data will be stored in the warehouse itself for easy access during the data mining process. There is, however, a cost associated with this storage (in terms of hardware, maintenance and performance) and this cost needs to be weighed up with the possible benefits. As the cost of on-line storage decreases, the storage of all transactional data in the warehouse becomes more feasible.

The transactional information can be archived direct from the source operational systems. Often this is performed as a matter of course for accounting or legal reasons. In these cases, it is still possible to access the data should the need arise in a future data mining project — however, the cost and complexity of the data mining project will increase dramatically. There is also a risk that the archived data will be mis-interpreted, or that it will be impossible to link it to the warehouse data due to unclear or incomplete metadata.

An alternative approach is to store transactional information in the warehouse for the immediate past, and then to archive directly from the warehouse. In this way, the transactional information is 'piped' through the warehouse and therefore archived in a compatible format. This will greatly increase the efficiency of any future data mining project.

4.2 Coverage

Data mining projects often require data covering many areas of the business and so it is important that the data stored in the warehouse has a good coverage of the business units. It is also important to consider external data — this defines the environment within which the business is operating. Examples of external data which might be required are: census data, weather data, competitor data etc.

In some cases, the external data may be considered of sufficient importance strategically that it is worth maintaining on the data warehouse as a matter of course — for example competitor data may fall in this category. In other cases, external data may only have

relevance for a particular data mining exercise and should be bought in on a case-by-case basis. One useful requirement for a corporate warehouse metadata repository is to have a shopping list for external data to aid the data mining analyst in locating the external data necessary for a particular data mining exercise.

Once external data has been used for a data mining exercise, there rests the issue of whether to maintain the external data on the warehouse, or whether to delete or archive the external data. The costs of storing the data have to be assessed against the likelihood of the data being re-used in the future. In particular, it is important to guard against the build-up of dormant data in the warehouse — i.e. data which is surplus to requirements, is costly to maintain and impedes the user from accessing the data they are interested in. Dormant data is particularly dangerous if there is insufficient metadata with which to identify it.

4.3 Temporal issues

This section discusses issues concerned with the currency of data — i.e. how the data is interpreted in the context of the business. Warehouses contain dynamic data, typically transactional information, and 'static' data such as reference data. This reference data is not truly static, however. Customers change address, marital status, salary levels; product hierarchies change; outlets change etc. Treating the data as static can lead to errors when analysing data relating to a fixed time period. As described in Section 3, this is a common scenario for data mining.

As an example, consider the analysis of retail shopping habits based on customer address and store location data. If it is assumed that people move house on average once every 8 years, this leads to 1% of the customer base moving house every month. Thus, if a data mining exercise carried out in January 1999 analyses data relating to Q1 1998, 10-12% of the customer address data is likely to be incorrect, if the current customer address data is used. This could cause potentially useful patterns to remain undiscovered, or produce spurious, noisy, patterns to mislead the analysis.

It is recommended that some form of versioning system is implemented in the warehouse for the reference data. This versioning system enables a point in the past to be recreated. The versioning system also allows "like for like" comparisons to be made between present and historical data.

The versioning data fields should include:

- Date when information became relevant.
- Date when information ceased to be relevant.
- Flag to signify currency (optional, for query performance).

The versioning system should be implemented when the reference data is loaded, or updated on the warehouse system. The system is easiest to implement if change data only is passed to the warehouse from the operational systems. However, a system can be used whereby the incoming data is compared with the previous data on the warehouse to create the change data. Either method may be used — this is generally dictated by the operational systems.

The costs of implementing this versioning system can be summarised as follows.

- **Extra disk space is required to store the versioned information.** For many tables the rate of change is not large, and so little extra disk space is required. For those

tables that do change frequently, the impact on total disk space is not necessarily an obstacle due to the relatively small data volumes of reference data compared to transactional data.

- **Human resource required to create and maintain versioning system.** There is an overhead required to create the scripts to perform the versioning and an overhead required to manage the execution of these scripts. However, provided the scripts are created in a modular, generic fashion, the scripts can be re-used once they have been created.
- **Computer resource required to execute versioning system.** The versioning system will add a component to the batch load. The amount of time required will depend on the type and extent of changes required.

It is possible for end users to access these columns directly when creating queries, and this is something that would appeal to power users. However, for standard end-users, the system described above is too complex to be used transparently. The functionality needs to be hidden using support in the end user tool.

5 Warehouse management

5.1 Metadata

The term "metadata" covers a number of concepts:

- the description of a table, or column in business terms;
- the location of a particular piece of data in the warehouse;
- information to guide users through the data;
- relationships in the data, at the business level (i.e. the business model);
- the owner or sponsor of the information;
- mappings to operational systems where the information came from ;
- data cleanliness description information such as missing data, estimated data, mis-classified data etc.
- mappings of derived data from the base data; and
- timestamps describing the time period to which the information relates.

The last three items in this list are related to data cleaning, data derivations and temporal issues, and are described in Sections 5.2, 5.3 and 4.3 respectively.

It is common for metadata repositories to contain the first concept — i.e. the description of a table or column in business terms. For example, the analyst can ask questions such as "What does field 'DD3' mean?" or "What is the scope of the 'margin' figure in table X?". However, for data mining purposes it is often useful to locate data for a hypothesis. The analyst might ask the question "Where can I find data on house prices?". It is therefore useful for the analyst to be able to use the warehouse metadata in order to locate the optimum data for the data mining exercise.

Metadata can be used in the data mining process to track when an interpretation of a column changes — for example new values may become applicable to a column at some point in time. Without suitable metadata to guide the analyst, this may result in

incorrect assumptions being made. As an example, consider a column "method of payment" in which there are three possible values "cash", "cheque" and "debit card" at the start of a three month data mining exercise. Half way through the third month, the business policy changes and credit cards become acceptable. This results in an extra value ("credit card") for the "method of payment" column. If this is not tracked in the metadata, there is a risk that this change will not be recognised and false assumptions about the use of credit cards may be reached.

Another use for metadata is to manage intermediate results that come from a data mining exercise. These intermediate results can be stored on the data warehouse to aid future analysis, but it is essential that metadata is used to make sense of these results.

5.2 Data cleaning

Data cleaning is an extremely important task for any data warehouse and all analysis can be potentially misled by dirty warehouse data. Data cleaning is mentioned here for completeness; however the subject is not discussed in depth.

An important point for data mining is to measure the cleanliness of the data on which mining is performed. Many cleaning issues may involve changes to source systems, which may not be possible within the timescales imposed by a data mining project. However, if the data cannot be cleaned, it must at least be possible to measure how dirty it is. It is recommended that metadata is used in these cases to enable clean samples to be obtained for data mining analysis.

It is important that data cleaning on the warehouse is not used to *change* data, but rather to *augment* the data. The real data can therefore be isolated and consistency with the operational systems is ensured. Physical methods of achieving this separation can be used — for example, storing cleaned data values in separate tables to the base data and using a view to access the overall cleaned data structure.

5.3 Derived and deduced data

This section describes the potential to enhance the data warehouse with data that has been derived or deduced from existing data. This derived and deduced data augments the base data and can be used by a number of business functions.

Derived data consists of transformations of the base data to produce new interpretations of the data, where the transformations do not introduce any doubt. It is important that transformations used in deriving data are contained in the warehouse metadata.

Deduced data may contain some assumptions, but can nevertheless be useful to the business. It is important that assumptions used in deducing data are contained in the warehouse metadata.

It is recommended that derived and deduced data is stored in separate tables to the base data, and joined when required using a join key. This approach will facilitate maintenance and ensure that the derived data can be easily separated from the base data.

6 Warehouse logical and physical structure

6.1 Logical structure

Operational databases generally use relational schemas, in third normal form, which are optimised for update efficiency and for storage efficiency. Data warehouses are optimised for storage; the principle difference between the requirements of the warehouse and the requirements of the operational system being that the warehouse has read-only access.

Warehouses may also be optimised for access. This involves de-normalising the relational form to produce the star or snowflake schemas. An alternative approach is to use a series of dependent data marts, optimised for access, which are fed from the data warehouse. This enables the warehouse itself to be optimised purely for storage.

During data mining, flexibility is a key criteria for all aspects of the system. This is a consequence of the unpredictable and iterative nature of the analysis — it is not possible to predict in advance the questions that will be asked, the exact data that will be required, the queries that will be executed etc.

Star and snowflake schemas are extremely efficient structures for the multi-dimensional model. However, the increase in efficiency is balanced by an increase in complexity, particularly with respect to the reference data. This complexity leads to a decrease in flexibility and it becomes difficult for the warehouse to be sufficiently dynamic to track the changing business. For data mining, flexibility is important and for this reason it is recommended that data is stored in normalised form in the data warehouse. Star or snowflake schemas can be used, but as an additional layer on top of the warehouse, or in a dependent data mart.

6.2 Physical structure

6.2.1 Architecture

This section discusses the physical architecture for a data mining solution. Three cases are considered.

1. Logic and data manipulation occur at the client. This implies that a high bandwidth is required when extracting data from the warehouse. Many warehouses are designed with high-bandwidth for data loading, but not for query access. This can lead to performance problems with this architecture.
2. Logic and data manipulation occur on a middle tier. The middle tier provides a single point of control which makes the system more flexible and manageable. The middle tier can also exploit high performance database or application servers, and so can lead to higher performance than the first case. The architecture requires a high bandwidth from the warehouse to the middle tier.
3. Logic and data manipulation occur on the database. This is SQL based data mining and is useful for some classes of mining algorithms. The data mining solution exploits the investment already made in the database platform.

In general, Cases 2 and 3 have the best fit with a warehouse infrastructure and these approaches can be combined, as demonstrated elsewhere in the CRITIKAL project [5]. The technical impact on the data warehouse is that computing resource needs to be

provided to run the data mining control and algorithms. High performance can be obtained through either:

- using a high-performance application server;
- using computing resource of the database machine as a high-performance application server; or
- using the database directly to obtain performance.

6.2.2 Capacity

Data mining projects often involve the creation of intermediate tables which are created during the analysis process. The "data mining table" described earlier is an example of this, but there may be additional tables. The number and size of these tables will vary depending on the nature of the data mining project.

It is important that there is sufficient disk capacity for these intermediate tables. In particular if it is required that the intermediate tables are maintained for future analysis, this must also be planned for.

6.2.3 Performance

This section considers the performance of two aspects of data mining: data transformation and data mining algorithms.

Data transformation

The production of the data mining table requires aggregation on the original atomic data. Typically there is a large quantity of this atomic data (100's of Gigabytes is not uncommon) and this operation is potentially extremely resource hungry. In particular, the aggregation often covers a large proportion of the original data and so efficient full table scans are required.

In order to get good performance in the transformation phase, the following techniques can be used:

- Data striping should be used to ensure that the data is spread over as many disks and disk controllers as necessary.
- A high performance disk subsystem should be used.
- Parallel methods should be used to access the data.
- High performance processors should be used.

These techniques are described further in [6].

Indexing techniques are often useful in operational and some warehousing environments. However, in data mining indexing is not so useful during the data transformation phase because the analysis often requires aggregation over the entire data set. This will not be the case when a sample is taken of the full data set for data mining; however this sample is taken only once during the iterative data mining process and is not a critical factor. Additionally, it is likely that the sampling criteria will be on fields that have not been previously identified as useful for indexing, or on fields where the sampling criteria is weakly selective. An example of the latter scenario is when extracting 3 months of data from 2 years of warehouse data. Here, the sample size is 12.5% of the entire data set — indexing is not a useful technique in these circumstances.

Data partitioning can be useful for this latter scenario. With data partitioning, data that is logically contiguous can be easily identified for extraction purposes. Data partitioning, by date/time, is particularly useful for transactional data.

Data mining algorithms
The algorithms themselves also make intensive use of full table scan operations. With many algorithms the selectivity criteria for subsets is fairly weak. For example, a decision tree algorithm will try to split the data in fairly equal sized bins at each node. Nodes that are deeper down in the tree will have selection criteria based on a number of fields. For example, a selection criteria may be "males under forty living outside London who have two children, one car and work in London". Each selection criteria is very weakly selective; however the final query will extract a relatively small subset of the population of the UK. Conventional indexing does not work in these circumstances. Bit-map indexing is a technique which can be applied to these circumstances; however, in many data mining algorithms the gains are still limited — for example, with decision tree algorithms advantage is only gained from extremely deep trees. These issues are discussed further in [7].

7 Conclusions

The level of detail to which data is stored in the warehouse has a critical effect on the degree of data mining that can be performed. It is recommended that atomic data is stored on the data warehouse, rather than the use of pre-aggregated data. This provides the necessary flexibility which is required for data mining analysis.

Ideally (for data mining purposes) the atomic data will be stored in the warehouse itself for easy access during the data mining process. As the cost of on-line storage decreases, the storage of all transactional data in the warehouse becomes more feasible. An alternative approach is to store transactional information in the warehouse for the immediate past, and then to archive directly from the warehouse. In this way, the transactional information is 'piped' through the warehouse and therefore archived in a compatible format. This will greatly increase the efficiency of any future data mining project.

Warehouses contain dynamic data, typically transactional information, and 'static' data such as reference data. This reference data is not truly static, however — treating the data as static can lead to errors when data mining. It is recommended that some form of versioning system is implemented in the warehouse for the reference data. This versioning system enables a point in the past to be recreated. The versioning system also allows "like for like" comparisons to be made between present and historical data.

It is recommended that use is made of warehouse metadata in order to locate the optimum data for the data mining exercise. Metadata should be recorded in the warehouse to track when an interpretation of a column changes.

It is recommended that metadata is used to manage intermediate results that come from a data mining exercise. These intermediate results and the accompanying metadata can be stored on the data warehouse to aid future analysis.

It is recommended that derived and deduced data, either produced as a result of data mining or produced for data mining, are stored on the data warehouse. The derived and deduced data should be stored in separate tables to the base data, and joined when required using a join key. This approach will facilitate maintenance and ensure that the

derived data does not obscure the base data. Metadata should be used to document any transformations used and assumptions made in deriving or deducing data.

For data mining, flexibility is important and for this reason it is recommended that data is stored in normalised form in the data warehouse. Star or snowflake schemas can be used, but as an additional layer on top of the warehouse, or in a dependent data mart.

It is recommended that processing resource is provided to run the data mining control and algorithms. High performance can be obtained through either:

- using a high-performance application server;
- using computing resource of the database machine as a high-performance application server; or
- using the database directly to obtain performance.

Indexing is often not useful for data mining due to the unpredictable nature of the data mining process and the frequent need to aggregate over the entire data set. Data partitioning can be useful for efficient extraction of large contiguous sections of data, particularly date/time partitioned transactional data.

8 References

1. William H Inmon. Building the Data Warehouse. John Wiley & Sons, 1996.
2. Tilmann Barth. Guidelines for the data mining process. EC Project CRITIKAL Deliverable D2.3, 1998.
3. P Chapman, J Clinton, J H Hejlesen, R Kerber, T Khabaza, T Reinartz, R Wirth. The Current CRISP-DM Process Model for Data Mining. Published by EC Project CRISP-DM on http://www.ncr.dk/crisp/, March 1998.
4. Evaluation of alternative strategies for derived attributes and virtual tables. EC Project CRITIKAL Deliverable D5.1, 1998.
5. M J Addis, P J Allen and J B Kingdon. Prototype HP-TNES implementation report. EC Project CRITIKAL Deliverable D6.3, 1998.
6. P J Allen. Report of Development of a Parallel Database: Codes of Best Practice . EC Project DBINSPECTOR Deliverable D4.3.0.
7. M J Addis, K Bosley, R Rantzau, H Schwarz. SQL-based performance optimisations for rule induction and association rules discovery. EC Project CRITIKAL Additional Deliverable AD6.4, January 1999.