Dynamic Data Mart for Business Intelligence

E. Chang, W. Rahayu, M. Diallo, M. Machizaud

Dynamic Data Mart for Business Intelligence

E. Chang[1], W. Rahayu[2], M. Diallo[3], M. Machizaud[3]

[1]The University of New South Wales, NSW, Australia
Elizabeth.chang@unsw.edu.au
[2]La Trobe University, Bundoora, Vic, Australia
W.Rahayu@latrobe.edu.au
[3] Ecole Nationale Supérieure des Mines d'Albi, Albi, France
{mariam.diallo,matthieu.machizaud}@mines-albi.fr

Abstract. Companies today have several major issues while managing information. Many subsidiaries and departments have developed their own Data Management which has led to a multitude of Operational Databases and sometimes a multitude of Data Marts, policies and processes. Thus, these systems lack sustainability because they are not dynamic and not self-organizing, and so they do not adapt to the continuous needs arising from evolution that the companies experience. The Dynamic Data Mart architecture is built around 6 main functions, namely the 3Ms (Data Mining, Data Marshalling and Data Meshing) and the 3Rs (Recommendation, Reconciliation and Representation), which will address the aforementioned problems. Once the totality of the data have been loaded into a single Data Warehouse, the Dynamics Data Marts address these problems by mining the user's behavior and the user's decision making processes and continuously and automatically adapting the Data Mart to the needs of the users. Dynamic Data Marts create adapted dimensions, facts, data associations and views and then automatically find the ones that are not used anymore. These latter are then automatically dropped by the system, or can be presented to the IT manager if needed for validation of their removal.

Keywords . Dynamic Data Mart, Data Integration, Disparate Data Sources

1. Introduction

Current Enterprise Data Warehouses are systems which have been constructed by experts in order to provide answers in the best way possible to business oriented questions. These questions have been posed by middle and top end managers, who needed to answer these questions contemporaneously to the building of the Data Warehouse. But an important problem remains: companies and organizations are in constant evolution in order to adapt to the market and to the changing environment. Processes, therefore, have to change and in this context, Data Warehouses still remain the same and often fail to meet the expectation of business users. Changing them could be arduous, time consuming and have a very long time duration. At the same time, business users have an increasing need to access a large amount of Data very quickly

in order to make better decisions. This has led different subsidiaries and departments to develop their own information systems, and therefore a large number of Data Bases and Data Marts exists within organizations (Figure 1). Therefore, as the demand of accurate, organized and useful data is increasing, an effective Data Management system is needed, in order to have consistency within the company and allow every level of manager to have access to consistent and reliable data.

2. Key issues

Three key issues have been identified in current Data Warehouse management systems namely :

1. A multiplicity of Data Marts exist in many big companies and organizations.

The information is spread among a number of subsidiaries and departments. Each subsidiary and department has developed its own system in order to store and access the data. Each of them may have created their own Data Warehouses and Data marts, in order to provide answers to very specific questions, resulting in a multiplicity of procedures, policies and user interfaces to manage the same Business processes. This often leads to inconsistency which becomes a real problem when the organization needs to have a larger overview of the information and when they need to drill down into and roll up over the different subsidiaries and departments.

2. Existing Data Warehouses are not sustainable

After a few years of use, as it is frequently not possible to answer new business question, because of lack of malleability of these Data Warehouses. The structure of the Data Warehouse and of Data Mart needs to be changed by the Data Warehouse Manager in order to answer these new questions, and it can take a lot of time to see these changes done. This important lead time results in the inefficient use of the total amount of data that the company has, resulting in the diminution of the creativity and of the curiosity of executives, and in the high possibility of missing important information due to a lack of freedom in the access to data. Data Marts are therefore not sustainable because they do not adapt easily to the changes of business processes and of policies.

3. Existing Data Warehouses cannot be dynamic or self-organize

This lack of sustainability of the data mart leads to the necessity of managing them constantly, in order to have a Data mart which is continuously consistent with the Business Processes. This process is particularly time-consuming and needs continuous care of the system: the quantity of data is indeed always growing whereas the processes are always evolving inside of the company. Therefore, the Data Warehouse is a hindrance to the development of the company, because it is not as dynamic and self-organized as they should be in order to answer the constantly changing needs of its users.

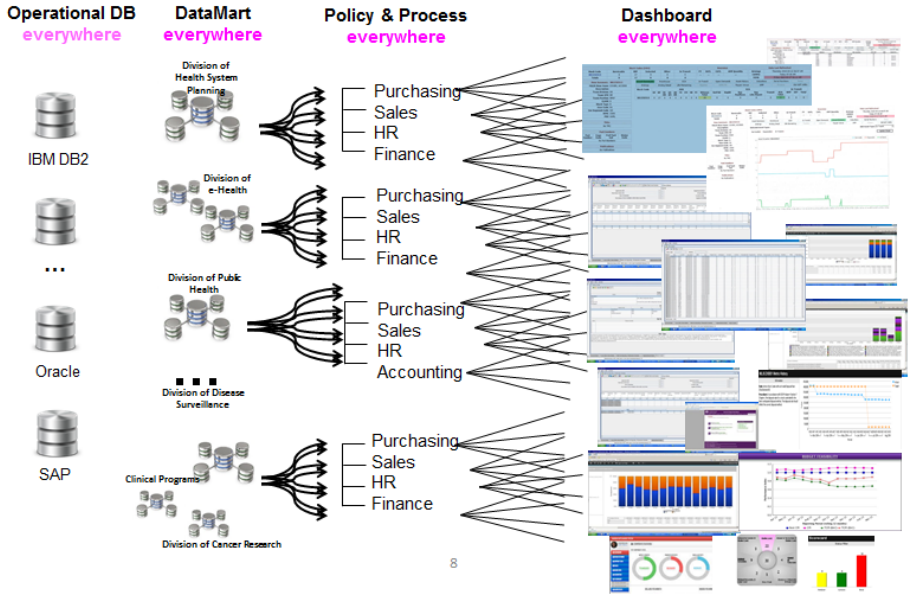

Figure 1: Traditional corporate Big Data Management and Database, Data Mart everywhere, leading to multiple overlapping policies, processes and Dashboards everywhere.

3. Existing Corporate Data Warehouse Implementation Architectures

Based on the latest Gartner (2015) study, the Corporate Data Warehouse data management architectures can be divided into the following 4 categories:

Traditional data warehouse: manages historical data coming from various structured sources. Data is mainly loaded through bulk and batch loading. It requires high capabilities for system availability, and administration and management, given the mixed workload capabilities for queries and user skills breakdown. This is a materialized or physical data warehouse.
Operational data warehouse: manages structured data that is loaded continuously in support of embedded analytics in applications, real-time data warehouses and operational data stores. Primarily supports reporting and automated queries, to support operational needs. This is a materialized or physical data warehouse/repository.
Logical data warehouse: manages data variety and volume for both structured and other content data types such as machine data, text documents, images and videos. Supports queries using data from sources other than the data warehouse DBMS alone. This combines materialized repository (for structured data) and non-materialised data warehouse where an enterprise integrated meta-model is defined that handles the

access to the individual data repository. The main components of this DW are the metadata, a data virtualization layer that can process data in their original source, and a distributed processing system. This is the most current architecture/methodology especially to support big data variety of data sets.

Context independent data warehouse: has the capability to establish "schema on read" approaches for new and even existing data values, variants of data form and new relationships. Also supports search, graph and other advanced capabilities for discovering new information models. There are no specific performance requirements and this option is favoured by advanced users such as data scientists or data miners, resulting in freeform queries across multiple data types.

Conventional data warehouses designs, which are primarily centered on relational databases have their own limitations. The two well-known models/schemas, star and snow flake due to their basis in relational models fail to adequately represent the semantics and operations of multi-dimensional data. There is always the problem of running complex (aggregate) queries on complex data. Also efficient execution of SQL queries is limited when drilling down in a data warehouse based on these models. The relatively new model/schema, star flake, which is the merger between the star and snowflake schema, manages to address some of these issues but fails to address all of them.

Later researchers, who adopted Object-Oriented techniques and Multidimensional data modelling for complex data, proposed many variations to the warehouse design. Many of the designs arise from the data mining On-Line Analytical Processing (OLAP) areas. Most of them concentrate on efficiency in query processing and data access rather than data semantics. Most of them were commercially unsuccessful as warehouse designs due to the “un-popularity” of Object-Oriented Database Management Systems (OODBMS) and limitations imposed by relational design constraints [1,2].

A data warehouse, primarily contains historical, consolidated data and should not lose its semantics at any point in time. But all methods suggested above are either non-semantically oriented or process oriented or a little bit of both. Almost all data warehouse models discussed above fail to capture the business side of the data warehouse. As a starting point, primary users of data warehouses are non-technical, middle and top end managers who have little or no knowledge of databases. No business rules can be captured inside the data warehouse. Most of the data warehouse designs are incapable of interpreting the business cost of data stored in them. As the business changes rapidly, the warehouses cannot or need complex query manipulation or re-design to accommodate the business changes. Though the data stored in them is historical, the information derived from them should accommodate the changes in the core business. At present the models above provide very little or no support for dynamic business information retrieval.

We are developing a methodology which utilizes current and new data warehouse design techniques to capture data semantics, business rules and business cost associated with each piece of information stored in or retrieved from the data warehouse. The new warehouse design will be based on the Object-Relational data model due to its ability to (1) capture both dynamic and static aspect of data ware houses, (2) utilize the growing O-R database market and (3) Semantic integration of data and systems.

4. Existing Data Warehouse Development Approaches

Several advanced approaches for dynamic data mart and virtualization for increased agility and reduced cost for Corporate Data Warehouse (CDW)/Business Intelligence (BI) applications include [6,7,16-18] and these are briefly discussed below:

1) Federated, multi-source data environment: A Data Virtualisation (DV) technology may access a data warehouse, in essence mirroring all of the existing consumable tables. DV can then extend this view to include other data to create a federated data capability. This can increase agility and reduce costs associated with physically moving data.

2) Spatial Temporal Data Warehouse: it contains geographical data sets, moving objects, the notion of timespan/valid time, historical tracking etc. It is important to make sure that the DW supports spatial and temporal notions. Spatial data warehouses (SDW) rely on extended multidimensional (MD) models in order to provide decision makers with appropriate structures to explore spatial data by using different analysis techniques such as OLAP (On-Line Analytical Processing). Current development approaches are focused on defining a unique and static spatial multidimensional (SMD) schema at the conceptual level over which all decision makers fulfil their current spatial information needs [10].

A conceptual multidimensional model includes spatial dimensions, spatial fact relationships, spatial hierarchies, spatial measures, and topological relationships and operations. This extension provides a new conceptual model called the MultiDimER. The GeoDWFrame framework has been proposed in [12], which is built on the star schema and used as a guide to design spatial dimensional schemas. This framework has two types of dimensions. The first type is geographical, which is categorized into primitive and composed dimensions that have at all levels only spatial data such as customers addresses, geo-references. The second type is a hybrid that is grouped into micro, macro, and joint that deals with spatial and conventional data such as customer's addresses, geo-references, product valid time/time span etc.

3) Virtual Data Marts: In this type data virtualization may augment CDW/BI by replacing some of the data marts with data virtualization objects (Views). A traditional DW might feed into virtual data marts, all within the DV platform. Again, such an approach can increase team agility and reduce costs associated with physically moving data. Such an architecture should be carefully vetted with a particular focus on performance.

4) Real-Time Data Warehouse: The traditional data warehouse was designed to store and analyse historical information on the assumption that data would be captured now and analysed later. System architectures focused on scaling relational data up with larger hardware and processing to an operations schedule based on clean data. Yet the velocity of how data is captured, processed, and used is increasing. Companies are using real-time data to change, build, or optimize their businesses as well as to sell, transact, and engage in dynamic, event-driven processes like market trading [6].

To enable **real-time data acquisition**, Oracle GoldenGate [19] uses log-based, real-time CDC (Change Data Capture and Delivery) capabilities to provide continuous

capture and delivery of the most recently changed data between OLTP systems and the data warehouse. CDC technologies identify and capture changes made to enterprise data sources, and then deliver those changes to target systems. The application offers transactional, real-time data capture, routing, transformations, and delivery, using the push approach. As soon as a new database transaction is committed at the source system, that data is immediately captured via the database transaction logs and moved to the data warehouse where it can drive enhanced, strategic, and operational BI capabilities.

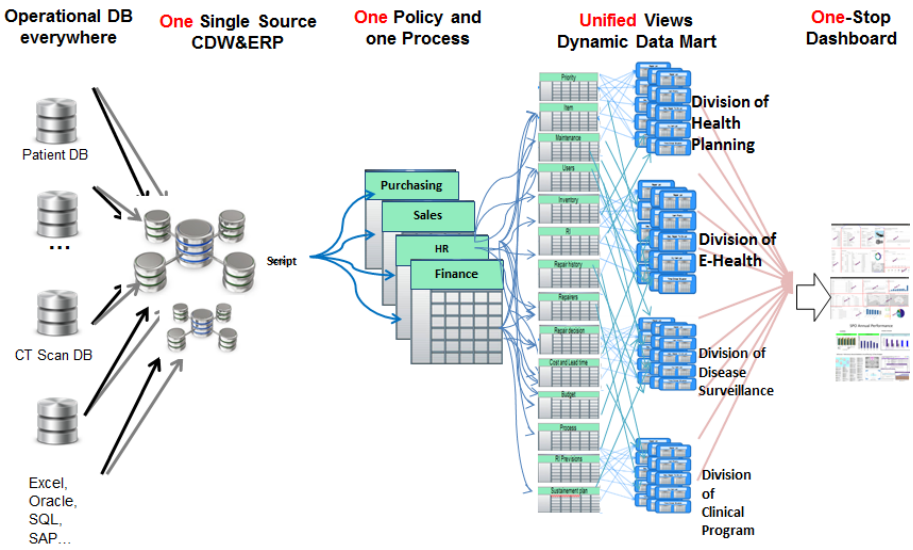

Figure 2 Dynamic Data Mart Architecture

5. New Dynamic Data Mart Implementation Framework

5.1 The Dynamic Data Mart Implementation Architecture

The dynamic data mart as shown in Figure 2 and 3 has two layers, namely:

- The dimensional layer and sub-dimensional layer, in Green colour
- The dynamic fact tables, shown in blue colour that will be represented as a view. The view is realized through the user interface, such as a pop-up window, or a form, or a graph, etc.

As there is a close connection between Capability, Acquisition, Sustainment and Disposal, it is important to have integrated dimensions. This is to avoid each department or subsidiary having to create their own data mart and pull in only the data they need, resulting in having data centres and data marts everywhere.

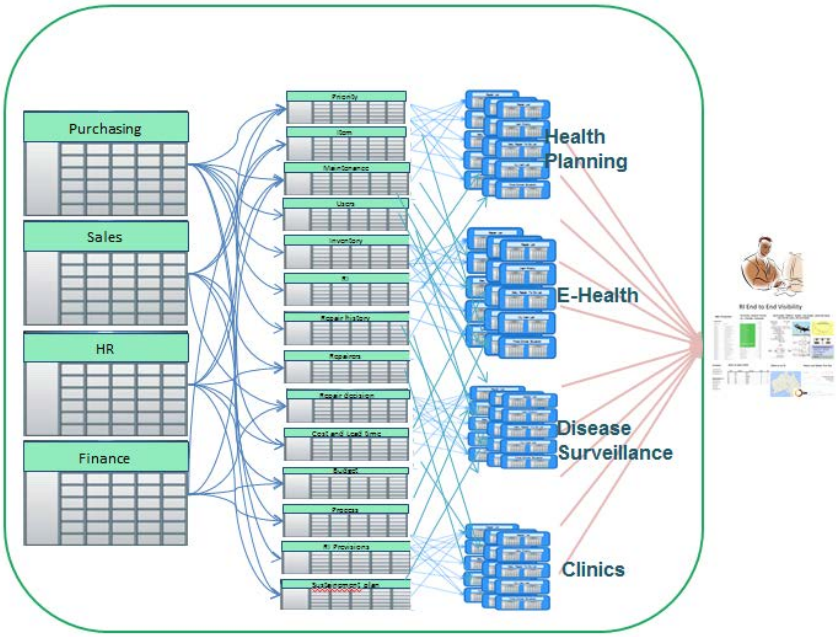

Figure 3 The Heart of Dynamic Data Mart

5.2 The Underlining Principles of the Dynamism

The Dynamic Data Mart Engine is a forward and backward loop that carries out 3M and 3R functions, as shown in Figure 4.

3M, namely:

Data Mining: mining the application log that mines the user's behaviours/user's decision making and usage rates of each view and window widgets clicks, providing usage rates.

Data Marshalling: for low usage rate views, we collect the data set, put them on a probation period, to see whether we can reuse.

Data Meshing: based on the data mining and data marshalling report, we create new views that potentially will attract the usage.

3R, namely:

Recommendation: Following up 3M, we provide recommendations to the user, in an analogous manner to how Amazon.com gives recommendations to people who have purchased a book by recommending them other similar books that other people have bought, that are likely to use the similar data set and make similar decisions, but this decision making is now recorded and reused.

Reconciliation: If the data is likely useful with high hit rate, but the view is not useful to finish a task, we reconcile all the window widgets and data sets to provide new window workflows or widgets workflows.

Representation: We then represent a new view to replace the old view to the user.

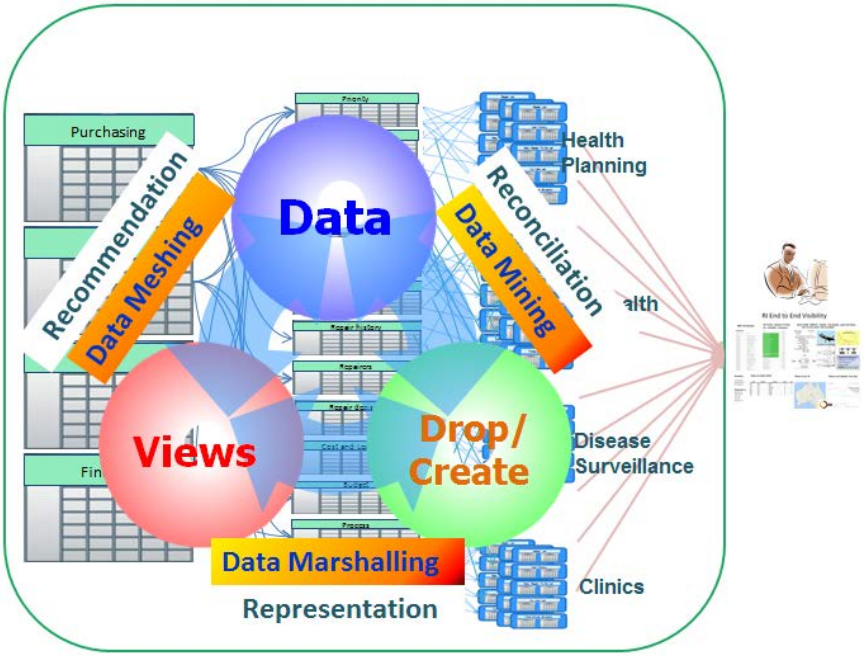

Figure 4. Dynamic Data Mart Engine

5.3 User Behaviour Mining, Log Mining and Usage Mining

We track the logs from a configured user's windows. The dashboard shows a number of areas, for each area we track the number sessions, the number of distinct users, peak concurrent sessions, cumulative duration of sessions and a user ratio.

Our 3M3R engine analyses peak concurrency events, solution adoption, decision making process, most active users (and candidates for Named Cals), and it drills down to individual session details (sessions tab) at each Window Area. It uses the trend chart on the concurrency tab to drill down to a minute level of detail.

The framework allows us to track how many times users open the model through the server log files and which user accessed the dashboard. By using Audit Logging we can track which objects and tabs are accessed by users and to perform this on the server the option Enable audit log has to be selected.

6 Dynamic Data Mart Implementation

6.1 Staging

STAGE 1 : CDW data dumps / Data Replication
A Corporate Data warehouse system contains various information system's database dumps. Currently, these database dumps are not consumed by any software application but for the dynamic DM technology we are going to use them to build a data warehouse. It is important to note that these database dumps are developed using various technologies such as Microsoft SQL Server 2008, IBM DB2, Excel Files Text files etc.

STAGE 2 : Data Progression
The data stored in the database dumps in CDW should be extracted, cleansed to remove inconsistencies and fill gaps, and integrated to merge heterogeneous sources into one common schema. In order to achieve this, the first step is different databases integration by defining global schema and storing this in the Meta-Data Repository. In a multiple database system, a global schema created by integrating schemas of the individual databases that provides a uniform interface and high level location transparency for the users to retrieve data. Once the global schema is established, the next step is to use Extraction, Transformation, and Loading tools (ETL) to merge heterogeneous schemata, extract, transform, cleanse, validate, filter, and load source data into a data warehouse. Technologically speaking, the staging area deals with the problems that are typical for different databases integration, such as inconsistent data management and incompatible data structures.

STAGE 3: From Corporate Data warehouse to Dynamic Data Marts (Dynamic DM)
As a result of ETL processes, the Information extracted and transformed will be located to one logically centralized single repository: a corporate data warehouse (CDW). During analysis and design of the data warehouse, various dimensions and facts tables will be identified according to faceplate requirements. These dimensions and facts tables will be loaded with the cleansed, transformed data created as a result of ETL process. The CDW will be used as a source for creating data marts such as Subsidiary 1…n, which partially replicate data warehouse contents and are designed for specific subsidiaries. Meta-data repositories store information on sources, access procedures, data staging, users, data mart schemata, and so on.

STAGE 4: User's Dashboard
Once the integrated data is efficiently and flexibly accessed to the user friendly Dynamic Data Mart and GUIs (Graphical User Interfaces), managers will be able to analyse items listed according to priorities, they will be able to dynamically analyse individual item/platform details, and will be able to simulate business scenarios. The Dynamic DM will be equipped with descriptive, predictive and prescriptive analytics models that will help managers to better visualize the pool size, history, forecast lead times and asset life etc.

6.2 ELT: Extract, Load, and Transform for Dynamic DM

In order to create the Dynamic Data Mart, we start with ELT:

1. Extract the data from the operational databases and load them into a corporate database (dump database).

2. Transform the data from the dump database into data marts (SQL scripts or other type of scripting)
3. Define the fact measure and connection to dimensions in order to create the fact tables and views.
4. Apply BI service on the Views.

6.3 The Dimension Tables and Fact Tables

In Data warehousing, the dimension table contains the textual or descriptive attributes of the data. For example, Customer dimension will contain details about customer's name, address, phone number etc.

Dimensions are used to slice and dice the data i.e. filter and group the data. Dimensional table which also help you by looking at data with "By" attribute i.e. say if the Total sales of the company is $1 Million then using Customer dimension you can look at the Total sales "By" Customer or "By" Time. A dimension table has a primary key column also called Dimension ID/Dim Id that uniquely identifies each dimension row. The dimension table is associated with a fact table using this key.

Fact Table contains the measurable attributes of the data. It contains measurable data that can be analysed by Dimension tables. Fact tables contain the foreign keys of the associated Dimension tables (figure 5).

6.4 The Data Model to Represent Fact and Dimension

The data model is used to develop dynamic association between fact tables and dimensions and its sub-dimensions. This allows self-organized Business Intelligence by providing automated easy drill-down operation for the end-user with automated data association.

Data Association is a technology that is widely used in modern BI tools, that builds the relationship between the concepts, and between the entities or tables, or between data-sets.

We create the Dynamic Data Mart with Dynamic Dimensions, Dynamic Tables and Dynamic Views.

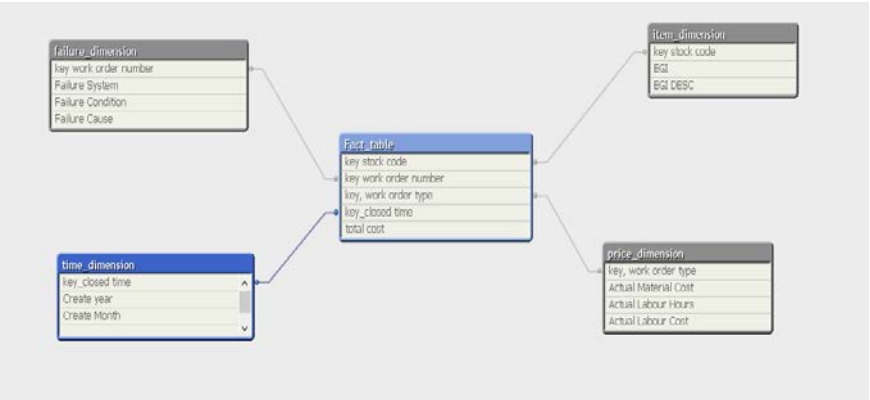

Figure 5 The Dynamic Data Model Sample

6.5 Dynamic create and drop views and data

Once we have the data on which dimensions are not used, we link this data to a new fact table named unused views and/or directly load the data to a dashboard in order to rank and visualize the views the least used.

Therefore we can choose to drop the views automatically if they are not used for more than 2 months (for example) or we let an IT manager decide if the views should be dropped. In the same way we can automatically create views knowing which data users accessed the most.

The automatic part would be created using an external application linked to our BI software or using a script inside our BI software if it is powerful enough. Using the data about usage (“name”; ”number of times used”), a simple formula targeting views can be used such as: with usage less than A=? and more than B=?. Then, we can create or delete views by linking the targeted views to the BI software (Figure 6).

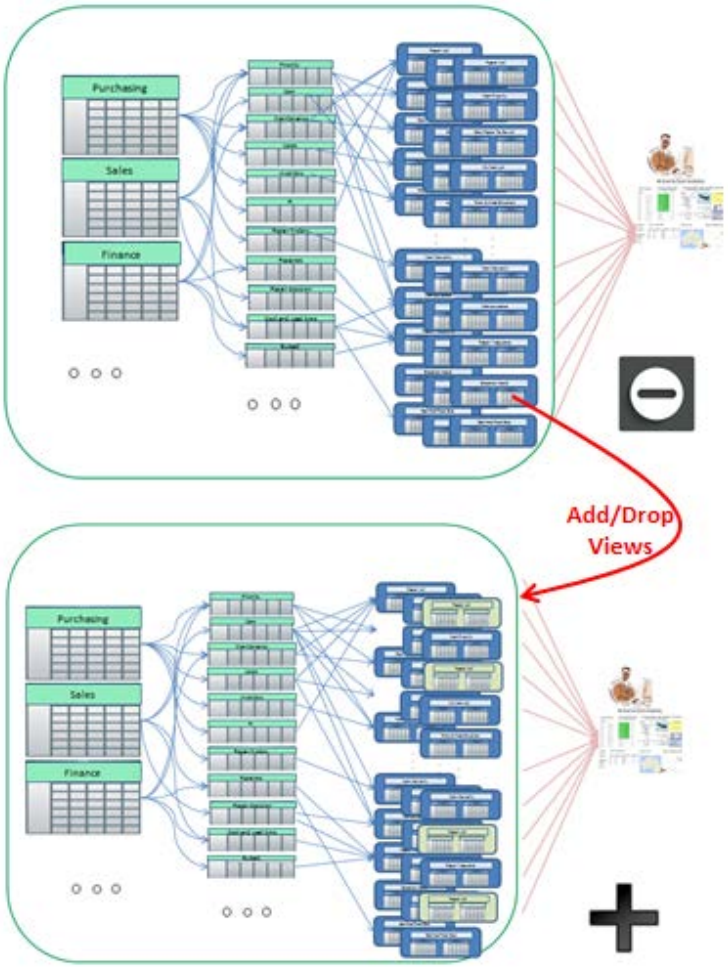

Figure 6: User Behaviour and usage mining, create/drop views

7. Application of Dynamic Data Mart

7.1 View Design and User Operation

We design the screens with partitions. All areas are linked together: we can know which area is the most used for changing all the dashboard (selection bookmark) but not which area is useful or useless. Indeed, an area could be good for visualizing information but not for selecting information and thus would appear as if it was never used. A solution could be to create a usage table where every time a data is changed on an area it adds one on "number of time used" (low number would mean it is not an important area). However, it would not be accurate since we do not know whether the user is using all the areas modified because most of them are linked together.

In addition, we will track down all the pop-ups, all the mouse clicks, and all the window widget. The steps involved include:

- Screen Design and partition for automated usability tracking
- Use of Scripts to create Dimensions and Sub-Dimensions
- Use of Scripts to create Facts Tables
- Use of scripts to Create data Mart
- Map Dynamic DM to User's Dashboard

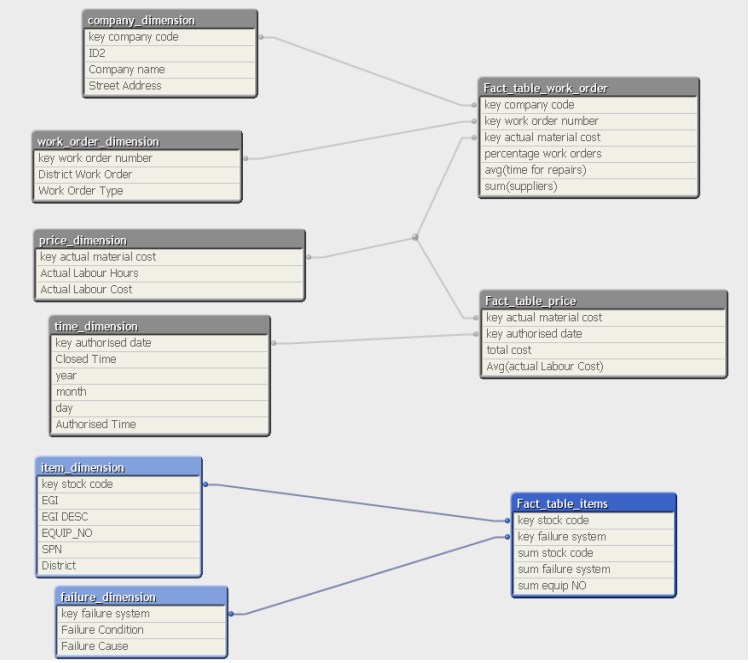

Figure 7 Example of use of Script for Create Data Mart Tables and Relationship

8 Conclusion

The paper examines the state of the art of Data Warehouses and how they are aligned with the company's processes. There is a strong need for Data Integration from multiple Data Sources and its effective use to answer questions posed by people in the business at different levels plus effective presentation in a form to which these business users can relate. A major weakness of current Data Warehouse and Data Mart approaches is the lack of adaptability so that evolution of the company's processes and Data requirements as well as new Business Questions can be addressed. This paper present an approach based on Dynamic Data Mart to overcome these weaknesses.

References

1. J. Wenny Rahayu, Elizabeth Chang, Tharam S. Dillon, David Taniar: Performance Evaluation of The Object-Relational transformation methodology. Data Knowledge Engineering 38(3): 265-300, 2001.
2. J. Wenny Rahayu, Elizabeth Chang, Tharam S. Dillon, David Taniar: A methodology for transforming inheritance relationships in an object-oriented conceptual model to relational tables. Information & Software Technology 42(8): 571-592, 2000.
3. Henry Chan, Raymond Lee, Tharam Dillon and Elizabeth Chang, E-commerce Principles and Practice, John Wiley and Sons, November 2001.
4. The Microsoft Modern Data Warehouse, Microsoft Corporation 2013/2014.
5. Data Integration Architectures for Operational Data Warehousing, Oracle, 2012
6. Data Virtualisation, Denodo Technologies, 2014 http://www.denodo.com/en/system/files/document-attachments/data_virtualization_goes_mainstream.pdf
7. Data Virtualisation, Data Source, 2013 http://datasourceconsulting.com/8-steps-data-virtualization/
8. Integrating Data Warehouse with Data Virtualisation, INTEL, 2013 http://www.intel.com.au/content/dam/www/public/us/en/documents/white-papers/virtualization-integrating-data-warehouses-for-bi-agility-paper.pdf
9. SAP HANA and Data Virtualisation, SAP Technology, 2012 http://stats.manticoretechnology.com/ImgHost/582/12917/2012/Resources/HANA-DV.pdf
10. Malinowski, E., and Zimányi, E., Logical Representation of a Conceptual Model for Spatial Data Warehouses. In GeoInformatica 11(4), p 431-457, Springer, 2007.
11. Nascimento Fidalgo, R., Times, V. C., da Silva, J., and da Fonseca de Souza, F., GeoDWFrame: A Framework for Guiding the Design of Geographical Dimensional Schemas. In Proceedings of the 6th International Conference on Data Warehousing and Knowledge Discovery (DaWaK), p. 26-37, Springer, 2004.
12. Octavio Glorio, Jose-Norberto Mazón, Irene Garrigós, Juan Trujillo, A personalization process for spatial data warehouse development, Decision Support Systems, 2012.
13. N.Stefanovic, J. Han, and K.Koperski, "Object-Based Selective Materialization for Efficient Implementation of Spatial Data Cubes", IEEE Transactions on Knowledge and Data Engineering, 2000.
14. R.Kern, T. Stolarczyk, NT Nguyen, A formal framework for query decomposition and knowledge integration in data warehouse federations, Expert Systems Applications 40, 2013

15. M. Jarke, T. List, J. Koller, The Challenge of Process Data Warehousing, VLDB conference, 2000.
16. Gartner, Magic Quadrant Data Warehouse Data Management Solutions for Analytics, 2015.
17. Gartner, Magic Quadrant for Data Quality Tools, 2014.
18. ThoughtWeb, Logical Data Warehousing for Big Data, 2013.
19. Data Integration Architectures for Operational Data Warehousing, Oracle, 2012.