An Overview of Data Warehouse Design Approaches and Techniques[‡]

Alejandro Gutiérrez, Adriana Marotta

Instituto de Computación, Facultad de Ingenieria, Universidad de la República, Montevideo, Uruguay

October 2000

Abstract

A Data Warehouse (DW) is a database that stores information oriented to satisfy decision-making requests. It is a database with some particular features concerning the data it contains and its utilisation. The features of DWs cause the DW design process and strategies to be different from the ones for OLTP Systems. This work presents a brief description of different approaches and techniques that address the DW Design problem.

Keywords

Data Warehouse (DW), DW design, schema transformation, Multidimensional data models, Relational DW.

[‡] This work was supported by Comisión Sectorial de Investigación Científica from Universidad de la República, Montevideo, Uruguay

1. Introduction

A Data Warehouse (DW) is a database that stores information oriented to satisfy decision-making requests. A very frequent problem in enterprises is the impossibility for accessing to corporate, complete and integrated information of the enterprise that can satisfy decision-making requests. A paradox occurs: data exists but information cannot be obtained. In general, a DW is constructed with the goal of storing and providing all the relevant information that is generated along the different databases of an enterprise.

A DW is a database with particular features. Concerning the data it contains, it is the result of transformations, quality improvement and integration of data that comes from operational bases. Besides, it includes indicators that are derived from operational data and give it additional value. Concerning its utilisation, it is supposed to support complex queries (summarisation, aggregates, crossing of data), while its maintenance does not suppose transactional load. In addition, in a DW environment end users make queries directly against the DW through user-friendly query tools, instead of accessing information through reports generated by specialists.

Building and maintaining a DW need to solve problems of many different aspects. In this work we concentrate in DW design.

The features of DWs cause the DW design process and strategies to be different from the ones for OLTP[1] Systems [Kim96-1]. For example, in DW design, the existence of redundancy in data is admitted for improving performance of complex queries and it does not imply problems like data update anomalies, since data is not updated on-line (DWs' maintenance is performed by means of controlled batch loads). Another issue to be considered is that a DW design must take into account not only the DW requirements, but also the features and existing instances of the source databases.

The goal of this work is to provide a quick overview of approaches and techniques used for providing solutions to the DW design problem both in industrial and research communities.

This paper is organized as follows. Section 2 presents the data warehousing area. Section 3 focuses on the approaches and techniques for DW Design. A particular approach used in the design of traditional databases is mentioned in Section 4. Finally, Section 5 presents our conclusion and our current work on the design of relational DW.

2. An overview of Data Warehousing

DW is a very wide research area. It has many different sub-areas and it can be treated with different approaches. Some overviews of the research area are [Wid95][Wu97][Cha97].

[1] OLTP: On Line Transaction Processing

The global architecture of DW systems considered in most works is the one shown in **Figure 2.1**, although there is a variant that is proposed in [Inm96]: a DW architecture with a Operational Data Store (ODS). This architecture is shown in **Figure 2.2**.

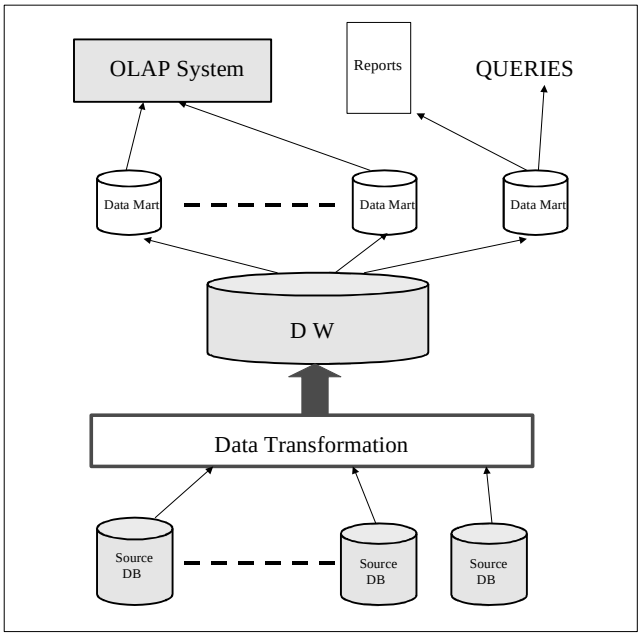

Figure 2.1

Source databases can be heterogeneous with respect to their data representation and to the data itself. Data integration is an important research area that treat this problem. Some publications that concentrate on source heterogeneity are [Pap96][Lev96], which consider in particular the web sources. In many projects as H20, TSIMMIS, DWQ, strong attention is paid to data integration [Hull96][Hull97][Pap96][Cal99].

In order to translate heterogeneous data models to a common model, some authors propose the use of wrappers [Lab97][Tork97], which encapsulate data sources and mediate between them and the rest of the system.

Data transformation layer involves a wide range of transformations that have to be applied to source data, for example data quality control and data cleaning, data integration, and conversions that are necessary for adapting data to the DW structures.

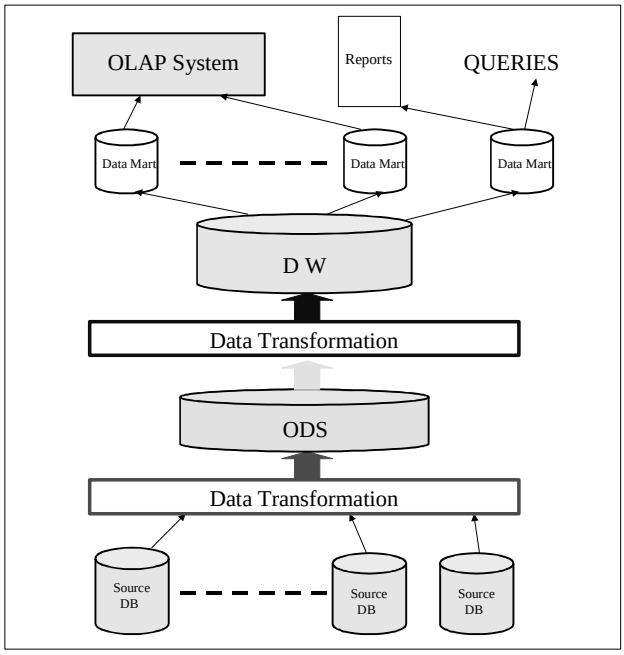

Figure 2.2

The ODS can be seen as an intermediate stage between the sources and the DW, although authors also propose that it can be used as a database for operational processing [Inm96][Kim96-2]. It contains integrated data, but this data is at detail level and it is only current data. Therefore we can think that with this architecture we are dividing the transformation work into two phases: in the first phase the main task is integration, and in the second phase the rest of the data transformation work is done.

Data Marts are proposed as logical subsets of the complete DW [Kim98]. They should be consistent in their data representation in order to assure DW robustness.

Concerning OLAP[2] [Tho97] systems, industry community has developed the main prototypes. Research community has not concentrated so much in it rather focusing in the proposals of precise multidimensional data models.

The data models that are used for DWs are Multidimensional Model and Relational Model. At the conceptual design level there is no discrepancy in choosing a multidimensional data model, since DW requirements are in general managed with a multidimensional perspective. Some publications about multidimensional data models are [Agr97][Gol98][Hac97]. The database system where the DW is built can be a multidimensional or a relational one. When this system is relational the logical design can be done applying techniques of multidimensional modelling to relational databases, as the ones presented in [Kim96-1].

[2] OLAP: On Line Analytical Processing

The most used approach, in research community, for definition and management of the DW is the one of materialised views. In the WHIPS project [Lab97][Ham95][Wie96] they work mainly in definition and maintenance of the DW [Zhu95][Lab96] and view consistency [Zhu96-1][Zhu96-2]. In the H2O project [Zhou95] they propose the combination of materialised and virtual views and they focus on data integration [Hull96][Hull97]. A very recent proposal about materialised views is in [Theo99-1], where they address the problem of selecting the views to materialise. We comment more about this approach in next section.

3. DW Design

As we have shown in **Figure 2.1**, a DW may be used by an OLAP front-end or it may be queried directly by SQL statements.

We found in the literature, globally two different approaches for Relational DW design: one that applies *dimensional modelling* techniques, and another that bases mainly in the concept of *materialized views*.

Dimensional models represent data with a "cube" structure [Kim96-1], making more compatible logical data representation with OLAP data management. According to [Kor99], the objectives of dimensional modelling are: (i) to produce database structures that are easy for end-users to understand and write queries against, and (ii) to maximise the efficiency of queries. It achieves these objectives by minimising the number of tables and relationships between them. Normalized databases have some characteristics that are appropriate for OLTP systems, but not for DWs: (i) Its structure is not easy for end-users to understand and use. In OLTP systems this is not a problem because, usually end-users interact with the database through a layer of software. (ii) Data redundancy is minimised. This maximises efficiency of updates, but tends to penalise retrievals. Data redundancy is not a problem in DWs because data is not updated on-line.

The basic concepts of dimensional modelling are: *facts*, *dimensions* and *measures* [Bal98]. A *fact* is a collection of related data items, consisting of measures and context data. It typically represents business items or business transactions. A *dimension* is a collection of data that describe one business dimension. Dimensions determine the contextual background for the facts; they are the parameters over which we want to perform OLAP. A *measure* is a numeric attribute of a fact, representing the performance or behaviour of the business relative to the dimensions.

Considering Relational context, there are two basic models that are used in dimensional modelling: (i) *star model* and (ii) *snowflake model*. The *star model* is the basic structure for a dimensional model. It has one large central table (fact table) and a set of smaller tables (dimensions) arranged in a radial pattern around the central table. (We show an example in **Figure 2.3**). The *snowflake model* is the result of decomposing one or more of the dimensions. The many-to-one relationships among sets of attributes of a dimension can separate new dimension tables, forming a hierarchy. (**Figure 2.4** shows an example). The decomposed snowflake structure visualises the hierarchical structure of dimensions very well.

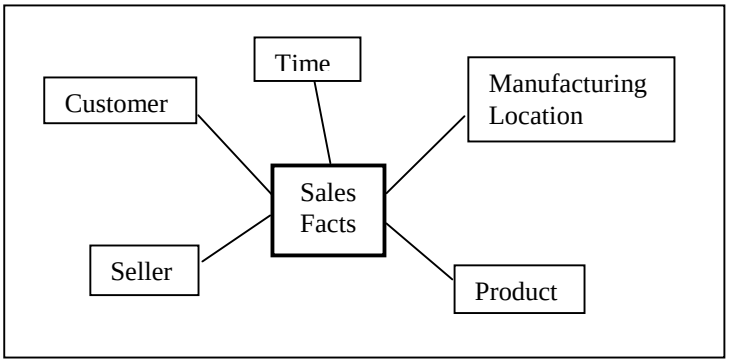

Figure 2.3

Other models that implement different design alternatives can be used. In [Kor99] they present a number of them, for example, *flat*, *terraced*, *star cluster*.

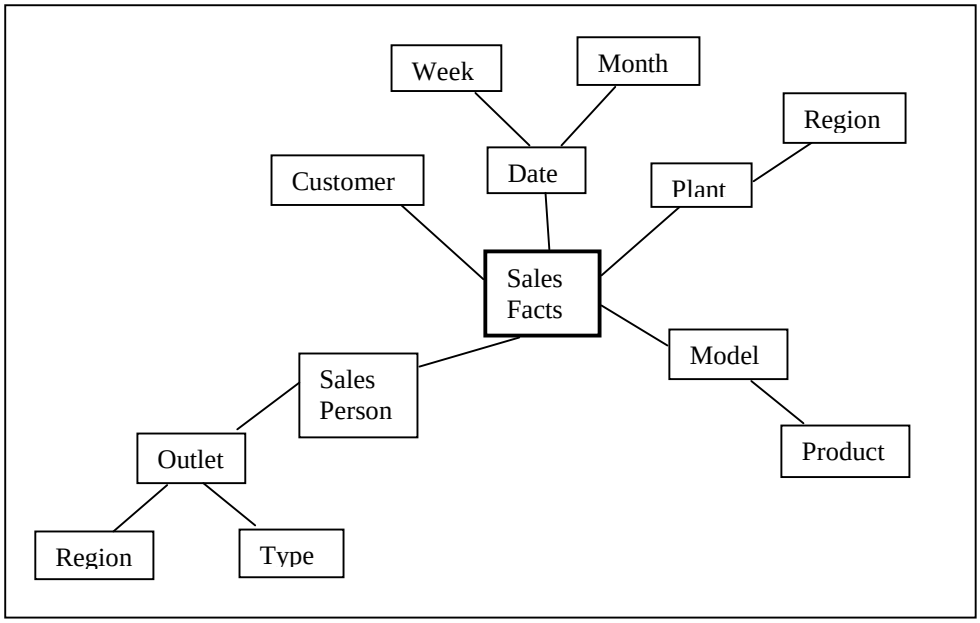

Figure 2.4

Practical design techniques and methods are proposed in [Kim96-1][Kim98][Kim96-3][Bal98], following mainly a star-schema approach. In [Ada98], authors also present concrete solutions for different target business. In [Sil97], they present DW models in a pattern-oriented approach, and propose techniques for converting a corporate logical data model into the DW model. In [Kor99] authors present a method for developing dimensional models from traditional Entity Relationship models.

In [Theo99-1] they focus on DW design, following the approach of *materialised views*. They address the problem of selecting a set of views to materialise in a DW taking into account: (i) the space allocated for materialisation, (ii) the ability of answering a set of queries (defined against the source relations) using exclusively these views, and (iii) the combined query evaluation and view maintenance cost. In this proposal they define a graph based on states and state transitions. They define a state as a set of views plus a set of queries, containing an associated cost. Transitions are generated when views or queries are changed. They demonstrate that there is always a path from an initial state to the minimal cost state. Some other publications about this approach are [Theo99-2][Lig99]. In [Theo99-3], working with the same model, they address the "Dynamic DW Design Problem", where basically, they determine which additional views have to be materialised when new queries have to be answered by the DW.

In general, existing work in DW design consists mainly of techniques for specific sub-models (as star or snowflake) and design patterns for specific domain areas. Although this work constitutes a precious knowledge base in DW design its practical application is not direct. In order to do it, designers must incorporate this knowledge, abstract the design rules and strategies, and then apply them in particular cases. Furthermore, this application would not be structured in well-defined design steps.

4. Schema transformation

Schema transformation primitives are a common used conceptual tool in the database area. In [Bat92], design primitives and strategies are presented as the building blocks of conceptual design methodologies. In [Hai91], they analyse the concept of schema transformation and generalise many of the proposed transformations in a conceptual schema design context. In [Sta90], database schema transformations are used and automated to perform schema evolution and reorganisation.

5. Conclusion

We present an overview of the data warehousing area and a brief description of data warehouse design approaches and techniques.

Concerning the data warehousing area en general, the most focused problems are data integration, extraction and transformation, data warehouse design and maintenance. Multidimensional data models are used for conceptual design of DW while variants of the relational data model are the usual data models used for the logical design. In particular, view materialization is the most used approach for DW management.

Considering the DW design problem, the usual technique to solve it is the proposal of algorithms to select the views to materialize in a DW. Schema transformation has been used for constructing and evolving traditional database schema. However, we have not find yet works trying to use schema transformation in a data warehouse context. We are working in this direction and propose a set of schema transformation primitives for relational DW design [Maro 99].

Our approach for DW design is not based on the materialisation of views. When using materialised views each desired relation of the schema must be able to be expressed in only one SQL query. Besides, we think that design process is easier and purer if it is done thinking only in the desired schema and not having to construct adequate SQL queries for obtaining the desired structures. In our work we clearly separate schema design from data loading, and we concentrate on schema design. According to our approach a DW schema can be designed transforming the source schema through a set of primitives and not depending on SQL expressiveness. Once the DW schema is designed, loading processes can be constructed.

Our intuition with respect to the set of primitives we designed is that they embed DW design techniques, covering the most common transformations that may be necessary for obtaining a DW schema from a source schema. In order to achieve this goal we base on the existing bibliography about DW design practical techniques and methods.

References

[Ada98] C. Adamson, M. Venerable. *Data Warehouse Design Solutions.* J. Wiley & Sons, Inc. 1998

[Agr97] R. Agrawal, A. Gupta, S. Sarawagi. *Modeling Multidimensional Databases.* ICDE 1997

[Bal98] C. Ballard. *Data Modeling Techniques for Data Warehousing.* SG24-2238-00. IBM Red Book. ISBN number 0738402451. 1998.

[Bat92] Batini, Ceri, Navathe. *Conceptual Database Design. An Entity-Relationship Approach.* The Benjamin/Cummings Publishing Company, Inc. 1992

[Cal99] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, R. Rosati. (DWQ project). *A Principled Approach to Data Integration and Reconciliation in Data Warehousing.* Proc. CAISE '99 Workshop on Design and Management of Data Warehouses (DMDW '99), 1999.

[Cha97] S. Chaudhuri, U. Dayal. *An overview of Data Warehousing and OLAP Technology.* SIGMOD Record 26(1). 1997.

[Gol98] M. Golfarelli, Stefano Rizzi. *A Methodological Framework for Data Warehouse Design.* DOLAP 1998.

[Hac97] M. S. Hacid, U. Sattler (DWQ project). *An Object-Centered Multi-dimensional Data Model with Hierarchically Structured Dimensions.* Proc. of the IEEE Knowledge and Data Engineering Workshop. 1997.

[Hai91] J. L. Hainaut. *Entity-Generating schema transformations for Entity-Relationship models.* ER 1991: 643 – 670.

[Ham95] J. Hammer, H. Garcia-Molina, J. Widom, W. Labio, Yue Zhuge. *The Stanford Data Warehousing Project.* Data Eng. Bulletin, 18(2), June 1995.

[Hull96] R. Hull, G. Zhou. *A Framework for Supporting Data Integration Using the Materialised and Virtual Approaches.* SIGMOD Conf., Montreal, 1996.

[Hull97] R. Hull. *Managing Semantic Heterogeneity in Databases: A Theoretical Perspective.* PODS 1997.

[Inm96] W. H. Inmon. *Building the Operational Data Store.* John Wiley & Sons Inc., 1996.

[Kim96-1] R. Kimball. *The Data Warehouse Toolkit.* J. Wiley & Sons, Inc. 1996

[Kim96-2] R. Kimball. *Dangerous Preconceptions.* The Data Warehouse Architect, DBMS Magazine, August 1996, URL: http://www.dbmsmag.com

[Kim96-3] R. Kimball. *Slowly Changing Dimensions.* The Data Warehouse Architect, DBMS Magazine, April 1996, URL: http://www.dbmsmag.com

[Kim98] R. Kimball. *The Data Warehouse Lifecycle Toolkit.* J. Wiley & Sons, Inc. 1998

[Kor99] M. A. R. Kortnik, D. L. Moody. *From Entities to Stars, Snowflakes, Clusters, Constellations and Galaxies: A Methodology for Data Warehouse Design.* 18th. International Conference on Conceptual Modelling. Industrial Track Proceedings. ER'99.

[Lab97] W. J. Labio, Y. Zhuge, J. N. Wiener, H. Gupta, H. Garcia-Molina, J. Widom. Stanford University. *The WHIPS Prototype for Data Warehouse Creation and Maintenance.* SIGMOD 1997.

[Lab96] W. Labio, H. Garcia-Molina. *Efficient Snapshot Differential Algorithms for Data Warehousing.* VLDB Conf., Bombay, 1996.

[Lev96] A. Y. Levy, A. Rajaraman, J. J. Ordille. *Querying Heterogeneous Information Sources Using Source Descriptions.* VLDB 1996.

[Lig99] S. Ligouditianos, T. Sellis, D. Theodoratos, Y. Vassiliou. (DWQ project). *Heuristic Algorithms for Designing a Data Warehouse with SPJ Views.* Proc. DaWaK '99, Florence, Italy.

[Maro99] A. Marotta. *A transformations based approach for designing Data Warehouses* Internal Report. InCo. Universidad de la República, Montevideo, Uruguay. 1999.

[Pap96] Y. Papakonstantinou, S. Abiteboul, H. Garcia-Molina. *Object Fusion in Mediator Systems.* VLDB 1996.

[Sil97] L. Silverston, W. H. Inmon, K. Graziano. *The Data Model Resource Book.* J. Wiley & Sons, Inc. 1997

[Sta90] B. Staudt Lerner, A. Nico Habermann. *Beyond Schema Evolution to Database Reorganization.* ECOOP/OOPSLA 1990 Proceedings.

[Theo99-1] D. Theodoratos, T. Sellis (DWQ project). *Designing Data Warehouses.* DKE '99

[Theo99-2] D. Theodoratos, S. Ligoudistianos, T. Sellis. (DWQ project). *Designing the Global Data Warehouse with SPJ Views.* Proc. CAISE '99, Heidelberg, Germany.

[Theo99-3] D. Theodoratos, T. Sellis. (DWQ project). *Dynamic Data Warehouse Design*. Proc. DaWaK '99, Florence, Italy

[Tho97] E. Thomsen. *OLAP Solutions. Building Multidimensional Information.* John Wiley & Sons, Inc., 1997.

[Tork97] M. Tork Roth, P. Schwarz. *Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources.* VLDB 1997.

[Wid95] J. Widom. *Research Problems in Data Warehousing.* Int'l Conf. On Info. And Knowledge Management (CIKM), November 1995.

[Wie96] J. L. Wiener, H. Gupta, W. J. Labio, Y. Zhuge, H. Garcia-Molina, J. Widom. *A System Prototype for Warehouse View Maintenance.* Workshop on Materialised Views: Techniques and Applications, June 1996.

[Wu97] Ming-Chuan Wu, Alejandro P. Buchmann. *Research Issues in Data Warehousing.* BTW German Database Conference, 1997.

[Zhou95] G. Zhou, R. Hull, R. King, J. Franchitti. *Data Integration and Warehousing Using H2O.* Data Eng. Bulletin, 18(2), 1995.

[Zhu95] Y. Zhuge, H. Garcia-Molina, J. Hammer, J. Widom. *View Maintenance in a Warehousing Environment.* SIGMOD Conf., San Jose, May 1995.

[Zhu96-1] Y. Zhuge, H. Garcia-Molina, J. Wiener. *The Strobe Algorithms for Multi-source Warehouse Consistency.* PDIS, Miami Beach, 1996.

[Zhu96-2] Y. Zhuge, H. Garcia-Molina, J. Wiener. *Consistency Algorithms for Multi-Source Warehouse View Maintenance.* Technical report, Stanford University, 1996.