Design and Management of Data Warehouses
Report on the DMDW'99 Workshop

http://sunsite.informatik.rwth-aachen.de/DMDW99/

Stella Gatziu* Manfred Jeusfeld† Martin Staudt‡ Yannis Vassiliou§

1 Introduction

The idea of building data warehouses as central data collections made available for decision support applications in a company is widely accepted. The concrete design and management of a data warehouse from a technical as well as from an organizational point of view, however, turns out to be far from trivial but requires sophisticated and time consuming efforts.

The DMDW workshop was held at the CAiSE'99 conference in Heidelberg on June 14-15, 1999. It had the intention to bring together practitioners and researchers to discuss the design and management of data warehouses. The various presentations gave a broad view on the data warehouse life cycle covering aspects relevant at design time, at build time and at run time. Overall, DMDW'99 was recognized as a success. The 30+ participants enjoyed the high quality program (acceptance rate of 50 percent) and had vivid discussions.

In this report, we review the presentations given at the DMDW workshop and present some open problems which we believe should be addressed by future research and whose solution could contribute to make data warehouse research more relevant to the practice.

2 Supporting data warehouse design

One of the most important tasks when designing a warehouse is to minimize the cost of answering queries because the warehouse is very large, queries are often ad-hoc and complex, and decision support applications require short response times. On the one hand, considering the warehouse itself as a set of views defined over the remote source data, its basic configuration, i.e. the available data and their interrelationship, clearly contributes to this. Furthermore, the processing costs of queries can be reduced by materializing view data for frequently asked queries. However, materializing all possible views may exceed the available storage space and imposes high cost for keeping the views up-to-date. The determination of the optimal collection of views in both cases is called the view selection problem. Three presentations at DMDW were devoted to this problem.

As a kind of preparatory task Michael Akinde and Michael Böhlen presented the construction of view graph structures to be used by view selection algorithms. The contribution of this work is the construction of such graphs in the presence of aggregation and grouping. The view graphs express how source data is materialized into views. The algorithm basically creates the search space of possible configurations of materialized views. Ad hoc rules are used to limit the size of the generated view graphs. One open problem is to combine a pruning strategy for the algorithm with specialized view selection algorithms.

A concrete approach to selecting views under quality requirements was proposed by Dimitri Theodoratos and Mokrane Bouzeghoub. They refine the view selection problem in order to include source availability constraints (how frequently can a data source be accessed for view maintenance) and currency constraints (how old can data elements in the data warehouse be). They show that the currency and source availability constraints can be restricted to simple views (on base relations). If the constraints satisfaction problem has no solution, the algorithm can identify the source relations which cause the constraint violation. If a solution exists, the algorithm computes the minimal update frequencies to achieve the desired data currency.

Randomized algorithms were found to be useful for choosing optimal query evaluation plans during query optimization. This idea can also be adopted for view selection as shown by Minsoo Lee and Joachim Hammer who employ a genetic algorithm to address the search space problem. The bits in a *genome* encode whether a view is materialized or not. Comparison to exhaustive search yields that the solutions delivered by the genetic algorithm are within 10% of the optimal solution. The algorithm is especially suited for data warehouses with a large number of views and with frequent changes to query definitions. Here, exhaustive search

*University of Zurich, Department of Computer Science, Winterthurerstrasse 190, 8057 Zurich, Switzerland; gatziu@ifi.unizh.ch

† Infolab, Tilburg University, Postbus 90153, 5000 LE Tilburg, The Netherlands; jeusfeld@kub.nl

‡ Swiss Life, Information Systems Research, P.O.Box, 8022 Zurich, Switzerland; martin.staudt@swisslife.ch

§National Technical University of Athens, Computer Science Division, Zographou 15773, Athens, Greece; yv@cs.ntua.gr

is intractable whereas the genetic algorithm delivers results within seconds.

While view selection is a technical problem usually solved in a relational context, the conceptual modelling of a data warehouse employs higher-level formalisms. Enrico Franconi and Ulrike Sattler observe that traditional conceptual data models (like ER diagrams) lack constructs for expressing aggregation. Since this feature is essential for data warehouses, they propose an extension to ER diagrams which allows to model aggregation over different dimensions. Multiple hierarchies of dimensions can be represented in parallel. The authors use the MD semantics of Cabbibo and Torlone to associate an interpretation to their models. Besides that, the models can be mapped to description logic expressions which enables reasoning on the conceptual level of warehouse design such as the detection of inconsistencies.

3 Practical aspects of warehouse design

Practical data warehouse design has to include the information demands existing in the data warehouse application context and also project management aspects. The usage of advanced commercial tools supporting certain aspects of data warehouse design becomes increasing important.

An issue arising before the actual conceptual modelling phase for a data warehouse takes place is information analysis. Han Schouten investigates in the analysis and design of data warehouses in a more general and abstract way without considering the warehouse as a set of views. He points out that data warehouse analysis concerns the analysis of the user needs for determining the required warehouse data and in particular the required derivations and the aggregation level. Data warehouse design consists in grouping derivable facts into data warehouse relations. Functional as well as so-called weak functional dependencies are considered for producing a suitable warehouse design. Existential graphs are proposed as a representation framework which offer a richer set of constructs to model cardinalities and constraints on entity attributes. It is argued that an expressive framework is needed to cover semantic properties like different versions of the subtype-supertype relationship.

While concentration in many software segments is still ongoing, the data warehouse tool arena is characterized by a broad disperse both in horizontal and vertical direction (many vendors of many special-purpose tools). Microsoft has included data warehouse specific amendments into their general OIM framework and is becoming a player in the ETL-tool market. Jens Otto Sørensen and Karl Alnor addressed data warehouse design from the tool angle. Their question was: Can a data warehouse be effectively designed using the Microsoft SQL Server (tm) and its DTS component? In their case study, they selected a relational schema for books and articles. A star schema was designed together with the data flow graphs specifying how data sources are fed into the data warehouse (the well-known *data pumps*). The data transfer is either implemented by SQL queries or by user-definable programs. The authors conclude that the available tools were sufficient and easy to use for designing the data warehouse and source integration. The case study was however restricted to the rather clean publications database delivered with the product.

Data warehousing consists of various processes to be executed at design time, at build time and at run time. The auditing within data warehouse projects was addressed by José Rodero, José Toval, and Mario Piattini with main emphasis on embedding the data warehouse into a company's organization. They identify the main activities: data source identification, data source integration, data storage, and analytical processing. For each of these activities, control objectives, metrics, and recommendations to cope with problems are presented. The framework follows the COBIT standard for information systems deployment and shares its business orientation.

4 Loading the data warehouse

The loading processes running on a data warehouse rely on complex specifications of parallel and interacting streams of operations on the data extracted from the sources.

Mokrane Bouzeghoub, Françoise Fabret and Maja Matulovic-Broqué propose to view data warehouse refreshment as a workflow application instead of a view materialization problem. Transferring data from sources to the data warehouse has to consider parameters like the availability of the sources. Moreover, data cleaning, integration, and customization are distinct processes which have to be carefully coordinated for the refreshment. The authors propose an event-driven workflow model to describe the interaction of the parallel processes. Different workflow reference models (called *scenarios*) are adapted to the specific requirements of a data warehouse project.

Diego Calvanese, Guiseppe De Giacomo, Maurizio Lenzerini, Daniele Nardi, and Ricardo Rosato focus on a declarative representation of the dependency between sources and warehouse allowing the generation of mediators for extraction and loading, rather than on the overall loading process. They propose a conceptual representation of data sources and their interrelationship by so-called correspondences. The representation has an interpretation in description logic (allowing reasoning about equivalence of schema concepts) and a Datalog interpretation (allowing to represent queries in terms of the concepts). Inter-schema constraints are ingredients for the reasoning method. Data conversion functions, e.g. for converting units, are represented as adornments to the Datalog representations. The concepts of the data sources are defined in terms of the enterprise model, i.e. the data sources are considered as views on an (imaginary) enterprise database. Whenever a new view is introduced in the data warehouse its specification is rewritten using the correspondences. The result is a mediator program which refers to the data sources and applies conversion, matching and reconciliation routines on and among them.

5 Data warehouse tuning

The data warehouse kernel, namely the (multidimensional) database requires special methods for optimization concerning cache management and indexing. The warehouse cache can contain precomputed data either already specified at design time (as materialized views) or based on the actual data warehouse usage.

Carsten Sapia investigates the design of interactive multidimensional data analysis tools (mostly OLAP systems) as an important part of the data warehouse design itself. The author discusses the modeling of user query behavior and its benefits based on the conviction that the OLAP design should be driven mainly by the user analysis requirements. He presents a mathematical model and a graphical notation for capturing knowledge about the typical multidimensional interaction patterns in OLAP systems, taking into account the session oriented, interactive and navigational nature of the user query behavior. The knowledge may then be used for pre-fetching data into a cache before the user has actually requested the data. The approach is currently be evaluated using log files of OLAP sessions.

Materialized views inside the data warehouse can potentially be reused during query evaluation. The modification of queries by substituting certain component by accesses to materialized views is called query rewriting. Sara Cohen, Werner Nutt, and Alexander Serebrenik propose algorithms for rewriting queries involving sum, count, min, and max aggregations. Existing algorithms do not address aggregation to its full generality. The authors extract from a data warehouse query its so-called *core* which selects all data elements subject to aggregation. For each rewriting of aggregate queries, soundness and completeness theorems are presented. The rewritings shall form the basis for a data warehouse query optimizer. Open questions are the handling of negation, functional dependencies, and queries with 'having' clauses.

Sophisticated index structures support the query processor but also introduce space vs. time trade-offs. Kiran Goyal, Krithi Ramamritham, Anindya Datta, and Helen Thomas identify access structures, esp. so-called *dataindexes*, as bottlenecks for performance of data warehouses. As indexes grow large, it becomes important to limit the number of disk accesses to locate an element in an index. The authors find that indexes expose a high degree of redundancy which makes them candidates for compression. Several compression algorithms were applied yielding compression ratios of up to 84%. It is concluded that compression should be considered as a new opportunity to increase the efficiency of query evaluation in a data warehouse.

Marcus Müller and Hans-J. Lenz presented the results of an extensive performance comparison of tree-based and bitmap indexes. Both are being used in data warehouse systems to support fast access to multi-dimensional data. The parameters of the comparison study were number of dimensions, number of tuples, number of different values for an attribute, number of attributes occurring in a query, size of a physical block etc. The authors show that bitmap index structures use modern disks better than traditional tree structures. The performance comparison exhibits bitmap indexes as the clear winner over tree indexes when the number of dimensions is rather high.

6 Meta data management

Meta data management has become increasingly important in data warehousing and related areas. The idea is to maintain information *about* the data warehouse which is required to install and evolve it. The meta databases of existing data warehouses mainly concentrate on logical and physical details, e.g. the table structures of the data sources and the timing of upload processes. So the question here is: Can and shall more information be handled by the meta database?

Peter Lehmann and Johann Jaszewski reported from experiences with Lawson-Mardon in introducing a data warehouse system. Lawson-Mardon is providing solutions for flexible product packaging for cosmetics, food, and pharmaceuticals. The data processing is done with a SAP R/3 installation. About 600 clients worldwide are supported by the system. It has been decided to view the data as a strategic resource and to make it available for decision making. The central statement is: *Only if data is linked to clear business terms, it obtains a meaning for the end user (of the data warehouse).* The authors propose to reconstruct the business terms from observations of the business practice. Afterwards, the statements are classified in a conceptual data model which can then be linked to the models of existing data sources. The authors reported that resolving weakly defined terms like "price" is one of the biggest problems in the reconstruction process. In fact, the reconstruction of business terms and their linkage to data sources consumed 70% of the total data warehouse installation time. This indicates that efficient methods for supporting this early phase are still missing. The authors mentioned data cleaning and the active use of meta data in the data analysis as further open problems.

Thomas Stöhr, Robert Müller and Erhard Rahm build upon a similar observation as the previous authors. Data sources in data warehousing have a very heterogeneous nature: COBOL files have to be processed as well as relational databases. A repository is proposed which manages meta data about technical characteristics (syntax of data, location of data) as well as semantic meta data necessary for report definition and for supporting the user's navigation in the data warehouse data. A review is given for available meta data systems. They can be classified as CASE repositories (allowing to model the business world with ER-like diagrams), OLAP repositories (representing multidimensional views on tables), data movement tools (representing data flows in a data warehouse), and plain repositories (representing meta data without a specific focus). The authors propose a UML

meta model which covers both data modeling and functional aspects of a data warehouse system. The main purpose of this meta model is to use it for more intelligent query formulation and navigation support. How to effectively do this is however still an open question.

Meta data about data warehouses also covers design processes. Christoph Quix presented a process model which is used to guide the data warehouse administrator in evolving the data warehouse, e.g., by integrating a new data source. The evolution steps are connected to quality factors like data warehouse interpretability. The process model is integrated into the meta database of a data warehouse which maintain conceptual, logical, and physical views on the objects and processes in a data warehouse system. Essentially, the evolution process model is a conceptual representation of a user manual for a warehouse administrator. By instantiating it, the meta database can record a trace for the evolution decisions done by the data warehouse administrator. This trace can be connected to actual quality measurements to assess the impact of the decisions.

7 Résumé

During the workshop we could again observe the known wide gap between research and practice in data warehousing. Actually, the term data warehouse was not invented by database researchers but by business driven consultants and practitioners. Nevertheless, a broad variety of data warehouse research projects have arisen during the last 5 years, addressing specific technical issues related to managing data warehouses. Although many solutions were developed for interesting subproblems like handling multidimensional data as typical requirement for data warehouses, view maintenance for aggregated data, data integration etc., combining these partial and often very abstract and formal solutions to an overall design methodology and warehousing strategy ist still left over to the practitioners who even cannot rely on existing commercial support in this respect. In addition, the influence of the research results on the commercial stream of data warehouse products is very limited due to the fact that the commercial data warehouse business has overrun and mostly ignored data warehouse research.

The practice is characterized by the need to embed a data warehouse into a company's business processes and therefore requires not only IT support but also input from the business administration side concerning project management and organizational aspects. The organizational structure of a company, its business goals, its decision support requirements and the market characteristics play an essential role for building a data warehouse.

The gap between data warehouse practice and research became obvious in particular during the discussion where questions like "How should I introduce a data warehouse in my company?" were actually posed but left unanswered. In this respect, the practical issues within the workshop program did not get the attention they deserved and the research perspective dominated. One possibility to reach a more symmetric constellation would be to bring commercial tool developers and data warehouse consultants into play who contribute ideas behind their tools and experiences within practical projects and who might be able to bridge both sides.

8 Research questions

Data warehousing is more than just building a central database in a company. Some interesting questions that arised during the workshop are the following:

Are different industries have specific requirements on the methods for data warehouse management? Can reference models for these industries be developed?

How can the introduction of a data warehouse into a company's organization be supported? Are there strategies which help to avoid faults? How comprehensive are these strategies?

How can we measure the success? Which are the stakeholders in a data warehouse? What are their goals? How is achievement of goals measured? Which design decisions contribute to the achievement of a given goal type?

What is the right combination of design and modeling languages? Are traditional data modeling languages sufficient? How should dynamic aspects like data uploading be modeled? What reasoning services are required and which are tractable?

What is the best data warehouse design for a given company? What are the analysis tasks? What are the properties of the data sources? Is there a design theory for data warehouses similar to the normal forms of relational databases?

How can we deal with evolution? How can new data sources be integrated with minimal negative maintenance cost for the dependent components?

How can a decision maker find out that the necessary information is included in the data warehouse? Is the vocabulary used by decision makers understandable to the administrators? How can a decision maker assess the quality of a decision depending on the quality of the information in the data warehouse?

This is surely an incomplete list of questions. Some of them are being addressed by current research, others are more or less neglected. DMDW will continue as a forum to discuss answers for these questions and, hopefully, find new questions as well.

References

[1] S. Gatziu, M.A. Jeusfeld, M. Staudt, Y. Vassiliou (eds.): *Design and Management of Data Warehouses*. Proc. Intl. Workshop DMDW'99 at CAiSE*99, Heidelberg, Germany, June 14-15, 1999, ONLINE: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-19/.