Preface

Nowadays the economy is characterized by fast and continuously changing markets and business opportunities. Therefore, in order to be successful, it is essential for an enterprise to make right business decisions and to make them fast. Business decisions are taken on the basis of analyses of the past and current condition of an enterprise as well as market analysis and predictions for the future. To this end, various business operational data collected during the lifetime of an enterprise are analyzed. Typically, operational data are stored within an enterprise in multiple data storage systems (subsystems) that are geographically distributed, are heterogeneous and autonomous.

The heterogeneity of data storage systems means that they come from different software vendors; they are implemented in different technologies (e.g., C, C++, .Net, Java, 4th generation programming languages); they offer different functionality (e.g., fully-functional databases, ODBC data sources, spreadsheets, Web pages, text files); they use different data models (e.g., relational, object-relational, object-oriented, semistructured) and different storage techniques; they are installed on different operating systems and use different communication protocols.

The autonomy of data storage systems implies that they are often independent from each other and remain under separate, independent control; that is, a local system's administrator can decide which local data are to be accessible from the outside of the system.

The management of an enterprise requires a comprehensive view of all aspects of a company, thus it requires access to all possible data of interest stored in multiple subsystems. However, an analysis of data stored in distributed, heterogeneous, and autonomous subsystems is likely to be difficult, slow, and inefficient. Therefore, the ability to integrate information from multiple data sources is crucial for today's business.

Data Warehouse and OLAP

One of the most important approaches to the integration of data sources is based on a *data warehouse* architecture. In this architecture, data coming from multiple external data sources (EDSs) are extracted, filtered, merged, and stored in a central repository, called a data warehouse (DW). Data are also enriched by historical and summary information. From a technological point of view, a data warehouse is a huge database from several hundred GB to several dozens of TB. Thanks to this architecture, users operate on a local, homogeneous, and centralized data repository that reduces access time to data. Moreover, a data warehouse is independent of EDSs that may be temporarily unavailable. However, a data warehouse has to be kept up to date with respect to the content of EDSs, by being periodically refreshed.

The content of a DW is analyzed by the so called online analytical processing (OLAP) applications for the purpose of discovering trends, patterns of behavior, and anomalies as well as for finding hidden dependencies between data. The outcomes of these analyses are then the basis for making various business decisions. The market analysis of demand and supply is one of important steps in taking strategic decisions in a company. Likewise, an analysis of the development and course of diseases as well as the impact of different medications on the course of illnesses is indispensable in order to choose the most efficient methods of treatment. Many other applications include, among others: stock market, banking, insurance, energy management, and science. Data warehouses and OLAP applications are core components of decision support systems.

Since the late 1980s, when the data warehouse technology developed, most of large and midsize companies worldwide have been building their own DWs into their information system infrastructures and have been successfully applying this technology in business. Major commercially available database management systems (e.g., Oracle9i/10g, IBM DB2 UDB, Sybase IQ, Computer Associates CleverPath OLAP, NCR Teradata Database, Hyperion Essbase OLAP Server, MS SQL Server, SAP Business Warehouse, SAS Enterprise BI Server) include the DW and OLAP technologies in their database engines. However, despite some substantial achievements in this technology, it still is and will be a very active research and technological field. The OLAPReport (2004) estimates that the total worldwide OLAP market constantly grew from less than $1 billion in 1994 to less than $5 billion in 2005, and it will grow up to $6 billion in 2007. The META Group's (currently Gartner) survey estimates that the OLAP market will be worth almost $10 billion in 2008 (EDWMarket, 2004). For these reasons, it is important to understand the core technological issues and challenges in the field of DW and OLAP.

Technological and Research Challenges

The size of a DW, high complexity of OLAP queries as well as the heterogeneous nature of integrated data pose serious research and technological challenges. Intensive research is conducted in several fields, that include, among others: schema design methodology and implementation models, data loading and DW refreshing techniques, efficient query processing, metadata management, and data quality issues (cf. Nguyen, Tjoa, & Trujillo, 2005).

A DW is designed for quick data retrieval by ad-hoc queries. These queries often compute aggregates based on other aggregates by rolling them up or they analyze details by drilling the aggregates down. Moreover, data are analyzed in the context of other data, for example, the monthly sales of products by particular shops. In order to support such kinds of analytical queries, a DW typically uses a multidimensional data model (Gyssens & Lakshmanan, 1997). In this model, facts representing elementary data being the subject of analysis are organized in n-dimensional spaces, called data cubes.

An n-dimensional data cube can be implemented either in MOLAP servers or in ROLAP servers. In the former case, a cube is stored either in a multidimensional array or in a hash table (e.g., SQL Server) or as the value of a binary large object (e.g., Oracle) or as another specialized data structure like Quad tree or K-D tree. In the latter case, a cube is implemented as the set of relational tables, some of them represent dimensions, and are called dimension tables, while others store values of measures, and are called fact tables. Two basic types of ROLAP schemas are used for an implementation of a cube, that is, a star schema and a snowflake schema (Chaudhuri & Dayal, 1997). The efficiency of executing OLAP queries strongly depends on an implementation data model and the type of the ROLAP schema used. Therefore, a lot of work is being spent on developing DW design methods (e.g., Adamson & Venerable, 1998; Kimball, Reeves, Ross, & Thornthwaite, 1998; Luján-Mora & Trujillo, 2004) and on modeling dimensions (e.g., Hurtado & Mendelzon, 2002; Letz, Henn & Vossen, 2002).

Having designed and implemented a DW one has to load it with data. The process of loading data into a DW is organized into several steps that include: (1) reading data from data sources, (2) transforming data into a common data model, (3) cleansing data in order to remove inconsistencies, duplicates, and null values, (4) integrating cleansed data into one set, (5) computing summaries, and (6) loading data into a DW. This process, called ETL (Extraction, Translation/Transformation, Loading), is executed by a software layer located between data sources and a DW (Kimball & Caserta, 2004; Simitsis, Vassiliadis, Terrovitis, & Skiadopoulos, 2005). The software includes: monitors that are responsible for detecting and reading changes in data sources, wrappers that are responsible for transforming a source data model into a common DW model as well as an integrator that is responsible for integrating data, cleansing them and loading them into a DW (Widom, 1995).

The initial loading into an empty DW reads all data of interest from EDSs and stores them in a DW. As the data sources are operational systems that are used everyday, their content changes frequently. As a consequence, the content of a DW becomes obsolete and has to be refreshed. A data warehouse is often implemented as the collection of materialized views, thus the problem of a DW refreshing transforms to the problem of maintaining and refreshing materialized views. This problem has been extensively investigated by the research community (Gupta & Mumick, 1999; Roussopoulos, 1998) and has resulted in multiple algorithms for refreshing materialized views. Typically, refreshing a materialized view is a costly task. In order to optimize it, multiple incremental refreshing techniques have been proposed. They can be categorized as refreshing with accessing data sources (e.g., Ceri & Widom, 1991) as well as self-maintenable refreshing (e.g., Samtani, Kumar, & Mohania, 1999). In the latter case, additional data structures are stored in a DW along with materialized views in order to eliminate the need of accessing data sources.

The process of refreshing a materialized view is usually executed concurrently with transactions on data sources and with user analytical queries. Such concurrent executions may result in inconsistent data stored in materialized views and erroneous results of analytical queries. Multiple solutions for avoiding these problems have been developed, that is, recomputing a view from scratch, applying compensation algorithms, maintaining versions of views, using additional data structures and transactions (e.g., Gupta & Mumick, 1999; Quass & Widom, 1997; Zhuge, Garcia-Molina & Wiener, 1996). Yet another problem is related to maintaining consistency of multiple dependent views during the process of their refreshment (e.g., Colby, Kawaguchi, Lieuwen, Mumick, & Ross, 1997; Zhuge, Wiener, & Garcia-Molina, 1997).

OLAP applications analyze data by means of complex queries ranging from a few to dozens operations of joining, filtering, grouping, and aggregating. Since these queries are very complex and they often read terabytes of data, their execution may take dozens of minutes, hours, or even days. Therefore, a key issue is the data warehouse efficiency. Well developed solutions to this problem are based on materialized views and query rewriting as well as on advanced index structures.

A challenging issue within the first solution concerns the selection of such a set of materialized views that: (1) will be used for optimizing the greatest possible number of the most expensive queries and (2) whose maintenance will not be costly. Several research works have addressed this problem and they have proposed multiple algorithms for selecting optimal sets of materialized views for a given query workload (e.g., de Sousa & Sampaio, 1999; Gupta, 1997; Theodoratos & Xu, 2004).

A specific characteristic of OLAP queries that typically join fact tables with multiple dimension tables as well as a specific distribution of values in fact tables requires different indexing schemes. Three kinds of indexes have been developed in order to optimize OLAP queries, namely, join indexes, bitmap indexes, and bitmap join indexes (e.g., Aouiche, Darmont, & Boussaïd, 2005; O'Neil & Graefe, 1995; Valduriez, 1987; Wu, Otoo, & Shoshani, 2004). The efficiency of executing OLAP queries

can also be increased by parallel processing and data partitioning techniques (e.g., Furtado, 2004; Rao, Zhang, Magiddo, & Lohman, 2002; Stöhr & Rahm, 2001).

The ETL and refreshing processes may insert erroneous or inconsistent data into a DW. As a consequence, user analyses will produce confusing or wrong results. That, in turn, may result in disastrous business decisions made by decision makers. For these reasons, research focused also on measuring and assuring data quality (e.g., Jarke, Jeusfeld, Quix, & Vassiliadis, 1998; Vassiliadis, Bouzeghoub, & Quix, 1999).

For a long period of time, the existing DW technologies have tacitly assumed that a DW is time invariant; that is, its schema and the structure of dimensions do not change during a DW lifetime. In practice, however, a DW structure changes as the result of the evolution of EDSs (Rundensteiner, Koeller, & Zhang, 2000), changes of the real world represented by a DW, new user requirements, as well as the creation of simulation environments, to list the most common cases. Several approaches to handling changes in DWs have been developed. They are categorized as supporting DW evolution (e.g., Blaschka, Sapia, & Höfling, 1999), temporal extensions (e.g., Hurtado, Mendelson, & Vaisman, 1999; Eder, Koncilia, & Morzy, 2002), simulation (e.g., Balmin, Papadimitriou, & Papakonstanitnou, 2000) as well as versioning (e.g., Body, Miquel, Bédard, & Tchounikine, 2002; Golfarelli, Lechtenbörger, Rizzi, & Vossen, 2004; Morzy & Wrembel, 2004).

In order to work properly and efficiently, all the above mentioned issues and techniques need to use metadata. Managing various types of metadata in DWs also has received a lot of attention resulting in widely accepted industry standard CWM (OMG, 2003; Vetterli, Vaduva, & Staudt, 2000), supported by major DW software vendors.

Despite the continuous development in the data warehousing technology that has lasted fore more than 20 years, it is still a very active area of research. Although most of the discussed research and technological achievements have been incorporated into various commercial database/data warehouse management systems, the discussed issues are still being investigated and the already implemented technologies are being further improved.

Further Development Directions

The already mature technologies discussed in the previous section are continuously being developed but new areas of research and new challenges appear on the scene. These new issues come from various novel information technologies applied to real business.

Nowadays, huge amounts of data are stored in various Web systems, typically in the XML format. These data are crucial for business and, therefore, there is a need

to analyze them in a similar way as in traditional DWs. This requirement led to several attempts to build data warehouses from Web data sources (e.g., Golfarelli, Rizzi, & Vrdoljak, 2001) and to provide OLAP functionality for XML documents (e.g., Nassis, Rajugan, Dillon, & Rahayu, 2005; Park, Han, & Song, 2005).

Advanced image processing technologies make the use of images and maps easier and more common, for example, Google Maps and NASA Earth Observing System Data and Information System (EOSDIS). In order to use information hidden in this kind of data a user needs a technology that combines the functionality of Geographical Information Systems with the functionality of data warehouses and OLAP. To this end, the so called Spatial OLAP systems are being developed (e.g., Stefanovic, Han, & Koperski, 2000).

Last but not least, our environment is becoming gradually filled with different kinds of sensors, for example, monitoring the intensity of traffic, controlling physical parameters of technological processes, and monitoring patients' vital signs. Sensors produce streams of data that have to be analyzed, often online. Stream data arrive also from other sources, for example, a click-stream form on the Internet, shares from a stock market, and transmission signals in telecommunications. Stream data are characterized by a continuous flow, requiring huge storage space. In order to reduce the space, historical data are stored as summaries or samples. Typically, incoming stream data need to be continuously analyzed. This leads to challenges in (1) querying streams online, (2) querying both historical summarized/sampled data and just incoming data, as well as (3) quickly accessing data, that is, indexing. Some substantial achievements have already been done in this area and some attempts have been made towards implementing DWs for stream data (Stream, 2006).

As it can be clearly observed, there are multiple kinds of data warehouses (relational, XML, spatial, stream) storing data of different formats and offering different functionality. Users of these technologies are often interested in combining data coming from multiple DWs. This requirement leads to the problem of integrating heterogeneous DWs (Torlone & Panella, 2005).

Book Objectives

The goal of this book is to provide an insight into important research and technological problems, solutions, and development trends in the field of data warehousing and OLAP. The content of the book encompasses important aspects of these technologies, from a DW designing and implementing, via data integration, loading, and DW refreshing, advanced query optimization techniques, to new areas of DW application. As such, the book:

- Provides the current state of the research and technology in the aforementioned domains,
- Constitutes a resource of possible solutions and technologies that can be applied when designing, implementing, and deploying a DW, and
- Serves as an up-to-date bibliography of published works for anyone interested in cutting-edge DW and OLAP issues.

Since the book covers a wide range of technical, technological, and research issues concerning DW and OLAP technologies, it is intended for data warehouse designers, administrators, programmers, and project managers. It offers them a better understanding of challenges, possible solutions, and advanced applications of these technologies. Moreover, technical aspects covered in the book will suit the contents of many DW courses offered at universities both in Europe and in the U.S. For this reason, the book can be a useful resource for students as well.

Structure of the Book

The body of the book consists of 13 chapters divided into four sections. Each of the sections addresses a particular research and technological area, namely:

- Modeling and designing,
- Loading and refreshing,
- Efficiency of analytical processing, and
- Advanced technologies and applications.

Section I addresses issues concerning one of the initial steps in the whole life cycle of a data warehouse, namely, requirements analysis, conceptual modeling, and designing. This section consists of three chapters.

Chapter I, *Conceptual Modeling Solutions for the Data Warehouse*, by Stefano Rizzi, focuses on a conceptual modeling that provides abstract representations of warehousing process, data structures, and architectures. The aim of conceptual modeling is to assure that a model is independent of an implementation. The author concentrates on a conceptual graphical model called the *dimensional fact model* (DFM) that was developed to support multidimensional modeling. The representation of the reality constructed using the DFM consists of the set of *fact schemata*. The basic concepts of the model include facts, measures, dimensions, and hierarchies. The DFM suits the variety of modeling situations that may be encountered in real projects of small to large complexity. The chapter provides a comprehensive set of

solutions for conceptual modeling according to the DFM and serves a DW designer as a practical guide for applying different modeling solutions. The chapter provides also a foundation for the rest of the book as it discusses fundamental concepts used in the DW technology, among others: multidimensional modeling; dimensions and their attributes; and shared, incomplete, recursive, and dynamic hierarchies.

Chapter II, *Handling Structural Heterogeneity in OLAP*, by Carlos A. Hurtado and Claudio Gutierrez, goes beyond the DFM model and it focuses on modeling dimensions that may have different structures; that is, they are heterogeneous. Such dimensions are created as the result of mixing multiple dimensions with different structures into a single dimension. In the chapter, the authors show how to incorporate structural heterogeneity in the design of OLAP models, explain why structural heterogeneity weakens aggregate navigation, survey different techniques to deal with heterogeneity, present a class of dimension integrity constraints to model structural heterogeneity, and demonstrate the practical application of dimension constraints to support aggregate navigation.

Chapter III, *Data Quality-Based Requirements Elicitation for Decision Support Systems*, by Alejandro Vaisman, presents a DW design method that supports complete and consistent elicitation of functional and nonfunctional requirements. Functional requirements take into consideration queries issued in applications, whereas nonfunctional requirements comprise data structures and data quality. The author argues that traditional design methods for requirements elicitation are inappropriate in the field of decision support systems, and presents the so called DSS-METRIQ method. The outcomes of the method are the set of requirement documents and the specification of the operational data sources that can satisfy user requirements. The chapter contains also the state-of-the-art in the field of requirements elicitation and design methods.

Section II addresses the problems related to loading data into a data warehouse and keeping the content of a DW up to date. This section is composed of two chapters.

Chapter IV, *Extraction, Transformation, and Loading Processes*, by Jovanka Adzic, Valter Fiore, and Luisella Sisto, is an experience report on the application of the ETL process to real world cases. The main focus of this chapter is on designing ETL processes for high data loading frequency, for large volumes of loaded data, and for complex data processing/transformations. In this context, the authors identify the most common critical issues and constraints that have an impact on designing the ETL processes. As the ETL design solution, the authors propose to apply a layered infrastructure. The infrastructure is composed of typically used functionalities and services in the ETL scenario and it is a basis for building various applications. The infrastructure includes, among others: DBMS access modules, file access modules, parallel read/write modules, and data processing modules. The authors also discuss and give practical suggestions on implementation issues including database partitioning options, parallel processing, and pipelining in the context of ETL.

Chapter V, *Data Warehouse Refreshment*, by Alkis Simitsis, Panos Vassiliadis, Spiros Skiadopoulos, and Timos Sellis, focuses on methods for designing efficient workflows of tasks within ETL processes. The features that make ETL challenging include, among others: huge data volumes, assuring the quality of data, assuring high performance, adapting workflows after changes in data sources, and changes in a data warehouse itself. As a response to these challenges, the authors propose a modeling approach/framework and its examplary application for the construction of ETL workflows. This approach is based on the life cycle of the ETL processes. The life cycle consists of four phases: reverse engineering and requirements collection; logical design; tuning and physical design; and software construction. The aim of the presented framework is to facilitate, manage, and optimize the design and implementation of the ETL workflows in order to create an optimal workflow. The framework supports all the phases of ETL design, from the initial design to a deployment stage and utilization, under continuous evolution of a data warehouse. The chapter contains also a comprehensive state of the art on commercially available tools and research achievements in the field of ETL.

Section III describes challenges and solutions to the problem of assuring the efficiency of analytical processing. Fundamental research and technological solutions to this problem include optimization techniques of star queries, indexing, partitioning, and clustering.

Chapter VI, *Advanced Ad Hoc Star Query Processing*, by Nikos Karayannidis, Aris Tsois, and Timos Sellis, focuses on efficient processing of OLAP queries. OLAP applications rely heavily on the so called star queries that join fact tables with multiple dimension tables. Reducing execution time of such joins is crucial for a DW performance. To this end, a new approach to fact table organization has been developed, called a hierarchical clustering. The hierarchical clustering allows clustering of fact data according to paths in dimension hierarchies. This clustering technique exploits path-based surrogate keys. In the context of hierarchical clustering, star query processing changes radically. In this chapter, the authors present a complete abstract processing plan that captures all the necessary steps in evaluating star queries over hierarchically clustered fact tables. Furthermore, the authors discuss issues on optimizing star queries within the context of the abstract processing plan and they define the abstract operations in terms of physical operations over the CUBE File data structure.

Chapter VII, *Bitmap Indices for Data Warehouses*, by Kurt Stockinger and Kesheng Wu, overviews the issues related to a special kind of index used for optimizing OLAP queries, namely, the bitmap index. Typically, bitmap indexes work well for attributes of low cardinality since the indexes are small for such atrributes. The higher cardinality of an indexed attribute, the larger size of a bitmap index. In order to reduce the sizes of bitmap indexes various techniques are used. This chapter overviews such techniques, namely, encoding, compression, and binning and it focuses on a particular compression technique called a word-aligned-hybrid compression.

Moreover, the authors present multiple experimental results comparing different encoding techniques and showing the characteristics of the word-aligned-hybrid compression. The results indicate that for high cardinality attributes compressed bitmap indexes also offer good index characteristics with respect to their sizes and query response times.

Chapter VIII, *Indexing in Data Warehouses: Bitmaps and Beyond*, by Karen C. Davis and Ashima Gupta, elaborates further on the issues presented in Chapter VII. In this chapter the authors focus on an alternative encoding technique that is based not only on values of indexed attributes but also on an additional knowledge derived from queries. This knowledge is used for constructing the so called property maps, each of which describes properties of indexed attributes. The characteristics of property maps with respect to query processing is the main contribution of this chapter. Additionally, the chapter contains a concise overview of different kinds of bitmap indexes.

Chapter IX, *Efficient and Robust Node-Partitioned Data Warehouses*, by Pedro Furtado, addresses the problem of running large data warehouses efficiently on low cost platforms. In order to achieve this goal, the author proposes to partition a data warehouse over multiple servers (nodes) in a network. Such a partitioning may cause other challenges in assuring data availability as well as distributed analytical query optimization. In this chapter the author shows how to use replicas for the purpose of designing a robust data warehouse partitioning strategy that guarantees efficient OLAP processing and data availability. The author also concentrates on data partitioning strategies as well as on efficient parallel join processing and query transformations.

Chapter X, *OLAP with a Database Cluster*, by Uwe Röhm, presents clustered data warehouse architecture as an alternative to the approach discussed in Chapter IX. The clustered architecture is based on a cluster of commercial off-the-shelf computers as hardware infrastructure that run off-the-shelf database management systems as transactional storage managers. In this architecture, the same data may be distributed over several nodes by using replication mechanisms. As a consequence, some replicas may be out of date. In order to handle queries on outdated data, the author proposes an approach to replication management, called freshness-aware scheduling, that introduces a new quality-of-service parameter. This parameter allows the specification of an explicit freshness limit in queries having an impact on replicas used in these queries. Based on the value of quality-of-service a query may be routed to the most appropriate node. Multiple query routing strategies and physical data design alternatives are also discussed in this chapter.

Section IV focuses on new domains for applying the data warehouse and OLAP technologies. These novel domains pose new challenges, among others in data modeling techniques, assuring analytical processing efficiency as well as in data storage/organization techniques.

Chapter XI, *Towards Integrating Data Warehousing with Data Mining Techniques*, by Rokia Missaoui, Ganaël Jatteau, Ameur Boujenoui, and Sami Naouali, addresses

problems and presents solutions for coupling data warehousing and data mining technologies. The work aims at developing an approach for flexible and efficient answer to data mining queries addressed either to a relational or a multidimensional database. The authors investigate the two following techniques. The first one exploits lattice based mining algorithms for generating frequent closed itemsets from multidimensional data. The second technique uses new operators, similar in spirit to the OLAP ones, in order to support data mining on demand. These new operators working on concept lattices include projection, selection, and assembly.

Chapter XII, *Temporal Semistructured Data Models and Data Warehouses*, by Carlo Combi and Barbara Oliboni, extends the application of the DW technology to semistructured data that may evolve in time. In order to store and analyze data of this kind, the authors propose a graph-based temporal semistructured data model. In this model, semistructured data are represented as a labeled graph with complex nodes representing abstract entities, simple nodes representing primitive values as well as with edges connecting nodes. Labels are associated with nodes and edges. Temporal aspects of data are handled by including valid time intervals in labels. In order to assure the consistency of this model, the authors propose two kinds of integrity constraints, namely, basic and domain-dependent ones. Basic constraints have to be satisfied by every graph, whereas domain-dependent constraints can be defined either for some specific nodes and edges or for the whole graph for a specific application domain.

Chapter XIII, *Spatial Online Analytical Processing (SOLAP): Concepts, Architectures, and Solutions from a Geomatics Engineering Perspective*, by Yvan Bédard, Sonia Rivest, and Marie-Josée Proulx, makes the reader familiar with challenges related to analyzing spatial data within the so called framework of spatial OLAP. The chapter outlines the particularities of spatial data and presents the state of the art of spatial OLAP applications. The main part of the chapter discusses the concepts, issues, challenges, and solutions related to spatial OLAP. This valuable discussion results from the experience of the authors with building a commercially available spatial OLAP system.

Robert Wrembel
Poznań, Poland

Christian Koncilia
Munich, Germany

June 2006

References

Adamson, C., & Venerable, M. (1998). *Data warehouse design solutions.* John Wiley & Sons.

Aouiche, K., Darmont, J., & Boussaïd, O. (2005). Automatic selection of bitmap join indexes in data warehouse. In A. Min Tjoa & J. Trujillo (Eds.), *International Conference on Data Warehousing and Knowledge Discovery* (LNCS 3589, pp. 64-73). Springer Verlag.

Balmin, A., Papadimitriou, T., & Papakonstanitnou, Y. (2000). Hypothetical queries in an OLAP environment. In A. El Abbadi, et al. (Eds.), *International Conference on Very Large Data Bases* (pp. 220-231). Morgan Kaufmann.

Blaschka, M., Sapia, C., & Höfling, G. (1999). On schema evolution in multidimensional databases. In M. K. Mohania & A. Min Tjoa (Eds.), *Data warehousing and knowledge discovery* (LNCS 1676, pp. 153-164). Springer-Verlag.

Body, M., Miquel, M., Bédard, Y., & Tchounikine, A. (2002). A multidimensional and multiversion structure for OLAP applications. In D. Theodoratos (Ed.), *ACM International Workshop on Data Warehousing and OLAP* (pp. 1-6). ACM Press.

Ceri, S., & Widom, J. (1991). Deriving production rules for incremental view maintenance. In G. M. Lohman et al. (Eds.), *International Conference on Very Large Data Bases* (pp. 577-589). Morgan Kaufmann.

Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. *SIGMOD Record, 26*(1), 65-74.

Colby, L. S., Kawaguchi, A., Lieuwen, D. F., Mumick, I. S., & Ross, K. A. (1997). Supporting multiple view maintenance policies. In J. Peckham (Ed.), *ACM SIGMOD International Conference on Management of Data* (pp. 405-416). ACM Press.

de Sousa, M. F., & Sampaio, M. C. (1999). Efficient materialization and use of views in data warehouses. *SIGMOD Record, 28*(1), 78-83.

Eder, J., Koncilia, C., & Morzy, T. (2002). The COMET metamodel for temporal data warehouses. In A. Banks Pidduck, et al. (Eds.), *International Conference CAiSE* (LNCS 2348, pp. 83-99). Springer-Verlag.

EDWMarket. (2004). *The evolving enterprise data warehouse market: Part 1*. Retrieved June 15, 2006, from http://www.teradata.com/t/pdf.aspx?a=83673&b=118524

Furtado, P. (2004). Workload-based placement and join processing in node-partitioned data warehouses. In Y. Kambayashi, et al. (Eds.), *International Conference on Data Warehousing and Knowledge Discovery* (LNCS 3181, pp. 38-47). Springer-Verlag.

Golfarelli, M., Lechtenbörger, J., Rizzi, S., & Vossen, G. (2004). Schema versioning in data warehouses. In S. Wang, et al. (Eds.), *Conceptual Modeling for Advanced Application Domains, ER 2004 Workshops* (LNCS 3289, pp. 415-428). Springer-Verlag.

Golfarelli, M., Rizzi, S., & Vrdoljak, B. (2001). Data warehouse design from XML sources. In J. Hammer (Ed.), *ACM International Workshop on Data Warehousing and OLAP* (pp. 40-47). ACM Press.

Gupta, H. (1997). Selection of views to materialise in a data warehouse. In F. N. Afrati & P. G. Kolaitis (Eds.), *Database theory: ICDT* (LNCS 1186, pp. 98-112). Springer-Verlag.

Gupta, A., & Mumick, I. S. (Eds.). (1999). *Materialized views: Techniques, implementations, and applications.* MIT Press.

Gyssens, M., & Lakshmanan, L. V. S. (1997). A foundation for multi-dimensional databases. In M. Jarke, et al. (Eds.), *International Conference on Very Large Data Bases* (pp. 106-115). Morgan Kaufmann Publishers.

Hurtado, C., & Mendelzon, A. (2002). OLAP dimension constraints. In L. Popa (Ed.), *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (pp. 169-179). ACM Press.

Hurtado, C. A., Mendelzon, A. O., & Vaisman, A. A. (1999). Maintaining data cubes under dimension updates. In *Proceedings of the International Conference on Data Engineering* (pp. 346-355). IEEE Computer Society Press.

Jarke, M., Jeusfeld, M. A., Quix, C., & Vassiliadis, P. (1998). Architecture and quality in data warehouses. In B. Pernici & C. Thanos (Eds.), *International Conference on Advanced Information Systems Engineering* (LNCS 1413, pp. 93-113). Springer-Verlag.

Kimball, R., & Caserta, J. (2004). *The data warehouse ETL tookit.* John Wiley & Sons.

Kimball, R., Reeves, L., Ross, M., & Thornthwaite, W. (1998). *The data warehouse lifecycle toolkit.* John Wiley & Sons.

Letz, C., Henn, E. T., & Vossen, G. (2002). Consistency in data warehouse dimensions. In M. A. Nascimento, et al. (Eds.), *International Database Engineering & Applications Symposium* (pp. 224-232). IEEE Computer Society Press.

Luján-Mora, S., & Trujillo, J. (2004). Physical modeling of data warehouses using UML. In K. Davis & M. Ronthaler (Eds.), *ACM International Workshop on Data Warehousing and OLAP* (pp. 48-57). ACM Press.

Morzy, T., & Wrembel, R. (2004). On querying versions of multiversion data warehouse. In K. Davis & M. Ronthaler (Eds.), *ACM International Workshop on Data Warehousing and OLAP* (pp. 92-101). ACM Press.

Nassis, V., Rajugan, R., Dillon, T. S., & Rahayu, J. W. (2005). A requirement engineering approach for designing XML-view driven, XML document warehous-

es. In *Proceedings of the International Computer Software and Applications Conference (COMPSAC)* (pp. 388-395). IEEE Computer Society Press.

Nguyen, T. M., Min Tjoa, A., & Trujillo, J. (2005). Data warehousing and knowledge discovery: A chronological view of research challenges. In A. Min Tjoa & J. Trujillo (Eds.), *Data warehousing and knowledge discovery* (LNCS 3589, pp. 530-535). Springer-Verlag.

OLAPReport. (2004). *Market share analysis.* Retrieved June 15, 2006, from http://www.olapreport.com/Market.htm

OMG (Object Management Group). (2003). *Common warehouse metamodel specification, v1.1.* Retrieved June 15, 2006, from http://www.omg.org/cgi-bin/doc?formal/03-03-02

O'Neil, P., & Graefe, G. (1995). Multi-table joins through bitmapped join indices. *SIGMOD Record, 24*(3), 8-11.

Park, B. K., Han, H., & Song I. Y. (2005). XML-OLAP: A multidimensional analysis framework for XML warehouses. In A. Min Tjoa & J. Trujillo (Eds.), *Data warehousing and knowledge discovery* (LNCS 3589, pp. 32-42). Springer-Verlag.

Quass, D., & Widom, J. (1997). On-line warehouse view maintenance. In J. Peckham (Ed.), *ACM SIGMOD International Conference on Management of Data* (pp. 393-404). ACM Press.

Rao, J., Zhang, C., Megiddo, N., & Lohman, G. (2002). Automating physical database design in a parallel database. In M. J. Franklin et al. (Eds.), *ACM SIGMOD International Conference on Management of Data* (pp. 558-569). ACM Press.

Roussopoulos, N. (1998). Materialized views and data warehouses. *SIGMOD Record, 27*(1), 21-26.

Rundensteiner, E., Koeller, A., & Zhang, X. (2000). Maintaining data warehouses over changing information sources. *Communications of the ACM, 43*(6), 57-62.

Samtani, S., Kumar, V., & Mohania, M. (1999). Self maintenance of multiple views in data warehousing. In *Proceedings of the ACM CIKM International Conference on Information and Knowledge Management* (pp. 292-299). ACM Press.

Simitsis, A., Vassiliadis, P., Terrovitis, M., & Skiadopoulos, S. (2005). Graph-based modeling of ETL activities with multi-level transformations and updates. In A. Min Tjoa & J. Trujillo (Eds.), *International Conference on Data Warehousing and Knowledge Discovery* (LNCS 3589, pp. 43-52). Springer-Verlag.

Stefanovic, N., Han, J., & Koperski, K. (2000). Object-based selective materialization for efficient implementation of spatial data cubes. *IEEE Transactions on Knowledge and Data Engineering, 12*(6), 938-958.

Stöhr, T., & Rahm, E. (2001). WARLOCK: A data allocation tool for parallel warehouses. In P. M. G. Apers et al. (Eds.), *International Conference on Very Large Data Bases* (pp. 721-722). Morgan Kaufmann.

Stream. (2006). *Stanford Stream Data Manager.* Retrieved June 15, 2006, from http://hake.stanford.edu/stream/

Torlone, T., & Panella, I. (2006). Design and development of a tool for integrating heterogeneous data warehouses. In A. Min Tjoa & J. Trujillo (Eds.), Data warehousing and knowledge discovery (LNCS 3589, pp. 105-114). Springer-Verlag.

Theodoratos, D., & Xu, W. (2004). Constructing search space for materialized view selection. In K. Davis & M. Ronthaler (Eds.), *ACM International Workshop on Data Warehousing and OLAP* (pp. 48-57). ACM Press.

Valduriez, P. (1987). Join indices. *ACM Transactions on Database Systems (TODS), 12*(2), 218-246.

Vassiliadis, P., Bouzeghoub, M., & Quix, C. (1999). Towards quality-oriented data warehouse usage and evolution. In M. Jarke & A. Oberweis (Eds.), *International Conference CAiSE* (LNCS 1626, pp. 164-179). Springer-Verlag.

Vetterli, T., Vaduva, A., & Staudt, M. (2000). Metadata standards for data warehousing: Open information model vs. common warehouse metadata. *SIGMOD Record, 29*(3), 68-75.

Widom, J. (1995). Research problems in data warehousing. In *Proceedings of the Fourth International Conference on Information and Knowledge Management* (pp. 25-30). ACM Press.

Wu, K., Otoo, E. J., & Shoshani, A. (2004). On the performance of bitmap indices for high cardinality attributes. In M. A. Nascimento et al. (Eds.), *International Conference on Very Large Data Bases* (pp. 24-35). Morgan Kaufmann.

Zhuge, Y., Garcia-Molina, H., & Wiener, J. L. (1996). The strobe algorithms for multi-source warehouse consistency. In *Proceedings of the Conference on Parallel and Distributed Information Systems* (pp. 146-157). IEEE Computer Society Press.

Zhuge, Y., Wiener J., Garcia-Molina, H. (1997): Multiple view consistency for data warehousing. In W. A. Gray & P. A. Larson (Eds.), *International Conference on Data Engineering* (pp. 289-300). IEEE Computer Society Press.