

Data Warehousing Systems: Foundations and Architectures

Il-Yeol Song

Drexel University, <http://www.ischool.drexel.edu/faculty/song/>

SYNONYMS

None

DEFINITION

A data warehouse (DW) is an integrated repository of data for supporting decision-making applications of an enterprise. The most widely cited definition of a DW is from Inmon [3] who states that “a data warehouse is a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management’s decisions.”

HISTORICAL BACKGROUND

DW systems have evolved from the needs of decision-making based on integrated data, rather than an individual data source. DW systems address the two primary needs of enterprises: data integration and decision support environments. During the 1980s, relational database technologies became popular. Many organizations built their mission-critical database systems using the relational database technologies. This trend proliferated many independent relational database systems in an enterprise. For example, different business lines in an enterprise built separate database systems at different geographical locations. These database systems improved the operational aspects of each business line significantly. Organizations, however, faced the needs of integrating the data which were distributed over different database systems and even the legacy database systems in order to create a central knowledge management repository. In addition, during the 1990s, organizations faced increasingly complex challenges in global environments. Organizations realized the need for decision support systems that can analyze historical data trends, generate sophisticated but easy-to-read reports, and react to changing business conditions in a rapid fashion. These needs resulted in the development of a new breed of database systems that can process complex decision-making queries against integrated, historical, atomic data. These new database systems are now commonly called data warehousing systems because they store a huge amount of data – much more than operational database systems – and they are kept for long periods of time. A data warehousing system these days provides an architectural framework for the flow of data from operational systems to decision-support environments. With the rapid advancement in recent computing technologies, organizations build data warehousing systems to improve business effectiveness and efficiency. In a modern business environment, a data warehousing system has emerged as a central component of an overall business intelligence solution in an enterprise.

SCIENTIFIC FUNDAMENTALS

OLTP vs. Data Warehousing Systems

Data warehousing systems contain many years of integrated historical data, ending up storing a huge amount of data. Directly storing the voluminous data in an operational database system and processing many complex decision queries would degrade the performance of daily transaction processing. Thus, DW systems are maintained separately from operational databases, known as online transaction processing (OLTP) systems. OLTP systems support daily business operations with updatable data. In contrast, data warehousing systems provide users with an environment for the decision-making process with read-only data. Therefore, DW systems need a query-centric view of data structures, access methods, implementation methods, and analysis methods. Table 1 highlights the major differences between OLTP systems and data warehousing systems.

Table 1: A comparison between OLTP and data warehousing systems

	OLTP	Data Warehouse & OLAP
Purpose	Daily business support. Transaction processing	Decision support Analytic processing
User	Data entry clerk, administrator, developer	Decision maker, executives
DB design	Application oriented	Subject-oriented
DB design model	ER model	Star, snowflake, Multidimensional model
Data structures	Normalized, Complex	Denormalized Simple
Data redundancy	Low	High
Data contents	Current, up-to-date operational data Atomic	Historical Atomic and summarized
Data integration	Isolated or limited integration	Integrated
Usage	Repetitive, Routine	Ad-hoc
Queries	Predictable, predefined Simple joins Optimized for small transactions	Unpredictable, Complex, long queries Optimized for complex queries
Update	Transactions constantly generate new data	Data is relatively static; Often refreshed weekly, daily
Access type	Read/update/delete/insert	Read/append mostly
Number of Records per access	Few	Many
Concurrency level	High	Low
Data retention	Usually less than a year	3-10 years or more
Response time	Subsecond to second	Seconds, minutes, worse

Systems requirements	Transaction throughput, Data consistency	Query throughput, Data accuracy
----------------------	---	------------------------------------

ROLAP & MOLAP

The data in a DW are usually organized in formats made for easy access and analysis in decision-making. The most widely used data model for DWs is called the dimensional model or the star schema [6]. A dimensional model consists of two types of entities—a fact table and many dimensions. A *fact* table stores transactional or factual data called *measures* that get analyzed. Examples of fact tables are *Order*, *Sale*, *Return*, and *Claim*. A dimension represents an axis that analyzes the fact data. Examples of dimensions are *Time*, *Customer*, *Product*, *Promotion*, *Store*, and *Market*. Since a DW contains time-variant data, the Time dimension is always included in dimensional schemas and the data in a fact table are organized by a unit of time. An extensive list of dimensions commonly found in DWs including those dimensions used in [1, 6] are presented in [4]. A typical structure of the dimensional model is illustrated in Figure 1 below.

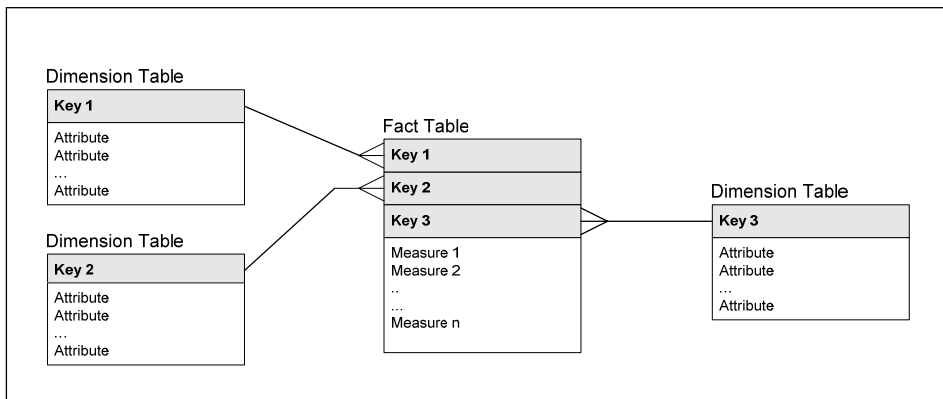


Figure 1: The typical structure of the star schema.

Syntactically, all the dimensions are connected with the fact table by one-to-many relationships. Thus, when a dimension has a many-to-many relationship with the fact table, a special technique such as an intersection table should be used. All the dimensions have a surrogate key, which establishes an identifying relationship with the fact table. In a star schema, all the dimensions are usually denormalized to simplify the query structure in order to minimize the number of joins. When dimensions are normalized into the third normal form, the schema is called a snowflake schema [6].

A dimensional model simplifies end-user query processing by simplifying the database structure with a few well-defined join paths. Conceptually, a dimensional model characterizes a business process with the fact table, the dimensions, and the measures involved in the business process. The dimensional model allows users of a DW to analyze the fact data from any combination of dimensions. The structure provides a multidimensional analysis space within a relational database.

Interactive data analysis of the data in a DW environment is called online analytic processing (OLAP). When the data in a dimensional model is stored in a relational database, the analysis is called relational online analytic processing (ROLAP). ROLAP engines extend SQL to support dimensional model schema and advanced OLAP functions.

DW data can also be stored in a specialized multidimensional structure called a data cube or a hypercube. Data analysis of the data stored in a data cube is called multidimensional OLAP (MOLAP). Compared with ROLAP engines, MOLAP engines are usually limited in data storage, but provide more efficient OLAP processing by taking advantage of the multidimensional data cube structure. A typical structure of a data cube is illustrated in Figure 2.

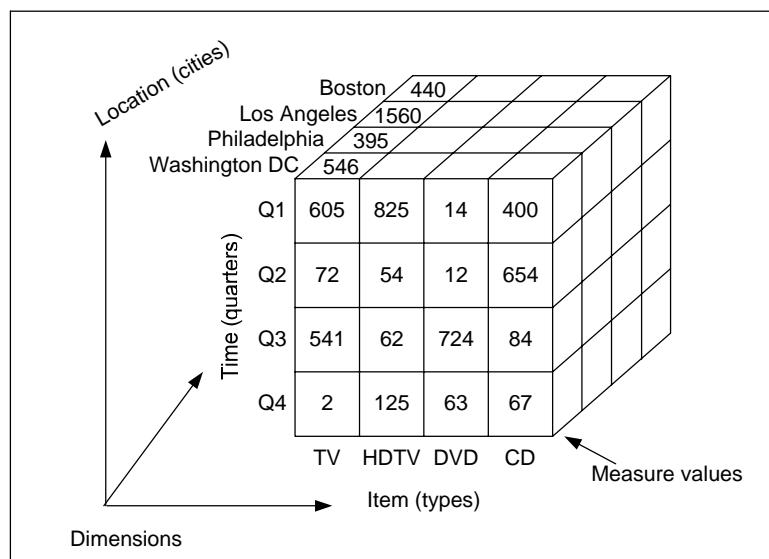


Figure 2: A three dimensional data cube having dimensions Time, Item, and Location for MOLAP.

Hybrid OLAP (HOLAP) servers take advantage of both ROLAP and MOLAP technologies. They usually store large volumes of detailed data in a ROLAP server and store aggregated data in a MOLAP server.

Data Warehousing Architecture

A data warehousing system is an environment that integrates diverse technologies into its infrastructure. As business data and analysis requirements change, data warehousing systems need to go through an evolution process. Thus, DW design and development must take growth and constant change into account to maintain a reliable and consistent architecture. A DW architecture defines an infrastructure by which components of DW

environments are organized. Figure 3 depicts the various components of a typical DW architecture that consists of five layers—data source systems, ETL management services, DW storage and metadata repository, data marts and OLAP engines, and front-end tools.

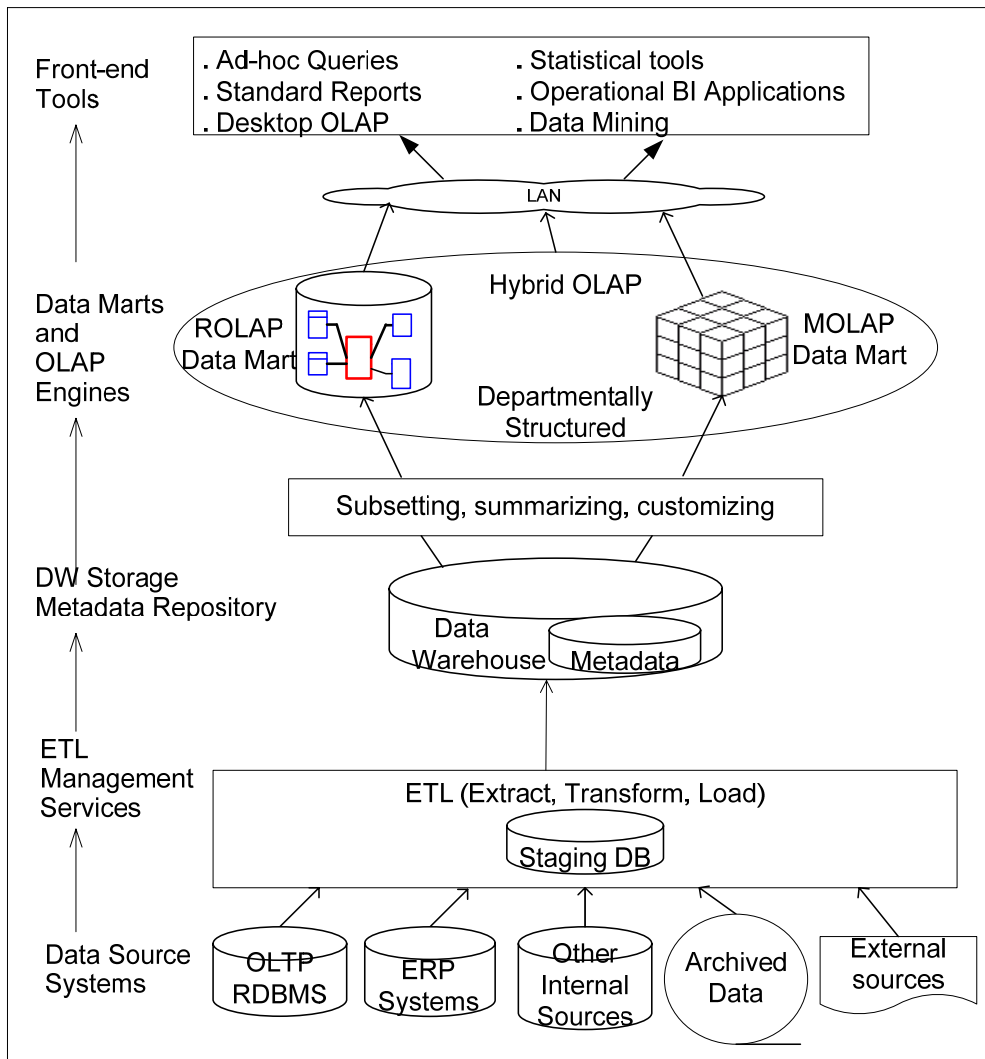


Figure 3: An enterprise data warehousing system architecture with ROLAP/MOLAP/Hybrid OLAP.

Data Source Systems

The data source system layer represents data sources that feed the data into the DW. An enterprise usually maintains many different databases or information systems to serve different OLTP functions. Since a DW integrates all the important data for the analysis requirements of an enterprise, it needs to integrate data from all disparate sources. Data could include structured data, event data, semi-structured data, and unstructured data. The primary source for data is usually operational OLTP databases. A DW may also

integrate data from other internal sources such as legacy databases, spreadsheets, archived storages, flat files, and XML files. Frequently, a DW system may also include any relevant data from external sources. Examples of such data are demographic data purchased from an information vendor to support sales and marketing analysis and standard reference data from the industry or the government. In order to analyze trends of data from a historical perspective, some archived data could also be selected. Thus, data warehousing systems usually end up with huge amounts of historical data.

These data are regularly fed into the second layer for processing. The interval between each feed could be monthly, weekly, daily, or even real-time, depending on the frequency of changes in the data and the importance of up-to-datedness of the data in the DW.

ETL Management Services

The second layer extracts the data from disparate data sources, transforms the data into a suitable format, and finally loads them to a DW. This process is known as ETL processing.

A DW does not need all the data from the data source systems. Instead, only those data that are necessary for data analysis for tactical and strategic decision-making processes are extracted. Since these data come from many different sources, they could come in heterogeneous formats. Because a DW contains integrated data, data need to be kept in a single standard format by removing syntactic and semantic variations from different data source systems. Thus, these data are standardized for the data model used in the DW in terms of data type, format, size, unit of data, encoding of values, and semantics. This process ensures that the warehouse provides a "single version of the truth" [3]. Only cleaned and conformed data are loaded into the DW. The storage required for ETL processing is called a staging database.

The ETL process is usually the most time-consuming phase in developing a data warehousing system [7]. It normally takes 60-80% of the whole development effort. Therefore, it is highly recommended that ETL tools and data cleansing tools be used to automate the ETL process and data loading.

Data Warehouse Storage and Metadata Repository

The third layer represents the enterprise DW and metadata repository. The enterprise DW contains all the extracted and standardized historical data at the atomic data level. A DW addresses the needs of cross-functional information requirements of an enterprise. The data will remain in the warehouse until they reach the limit specified in the retention strategy. After that period, the data are purged or archived.

Another component of this layer is the metadata repository. Metadata are data about the data. The repository contains information about the structures, operations, and contents of the warehouse. Metadata allows an organization to track, understand, and manage the

population and management of the warehouse. There are three types of metadata—business metadata, technical metadata, and process metadata [7]. *Business metadata* describe the contents of the DW in business terms for easy access and understanding. They include the meaning of the data, organizational rules, policies, and constraints on the data as well as descriptive names of attributes used in reports. They help users in finding specific information from the warehouse. *Technical metadata* define the DW objects such as tables, data types, partitions, and other storage structures, as well as ETL information such as the source systems, extraction frequency, and transformation rules. *Process metadata* describe events during ETL operations and query statistics such as begin time, end time, CPU seconds, disk reads, and rows processed. These data are valuable for monitoring and troubleshooting the warehouse.

Metadata management should be carefully planned, managed, and documented. OMG's Common Warehouse Metamodel [9] provides the metadata standard.

Data Mart and OLAP Engines

The fourth layer represents the data marts and OLAP engines. A data mart is a small-sized DW that contains a subset of the enterprise DW or a limited volume of aggregated data for the specific analysis needs of a business unit, rather than the needs of the whole enterprise. This definition implies three important features of a data mart, different from a DW system. First, the data for a data mart is fed from the enterprise DW when a separate enterprise DW exists. Second, a data mart could store lightly aggregated data for optimal analysis. Using aggregated data improves query response time. Third, a data mart contains limited data for the specific needs of a business unit. Conceptually, a data mart covers a business process or a group of related business processes of a business unit. Thus, in a fully-developed DW environment, end-users access data marts for daily analysis, rather than the enterprise DW.

An enterprise usually ends up having multiple data marts. Since the data to all data marts are fed from the enterprise DW, it is very important to maintain the consistency between a data mart and the DW as well as among data marts themselves. A way to maintain the consistency is to use the notion of conformed dimension. A *conformed dimension* is a standardized dimension or a master reference dimension that is shared across multiple data marts [6]. Using conformed dimensions allows an organization to avoid repeating the "silos of information" problem.

Data marts are usually implemented in one or more OLAP servers. OLAP engines allow business users to perform data analysis using one the underlying implementation model—ROLAP, MOLAP, or HOLAP.

Front-end Tools

The fifth layer represents the front-end tools. In this layer, end-users use various tools to explore the contents of the DW through data marts. Typical analyses include standard

report generations, ad-hoc queries, desktop OLAP analysis, CRM, operational business intelligence applications such as dashboards, and data mining.

Other DW Architectures

Figure 3 depicts the architecture of a typical data warehousing system with various possible components. The two primary paradigms for DW architectures are enterprise DW design in the top-down manner [3] and data mart design in the bottom-up manner [6]. A variety of architectures based on the two paradigms and other options exists [3, 6, 8, 10, 12]. In this section, seven different architectures are outlined. Figures 4-9 illustrate those architectures.

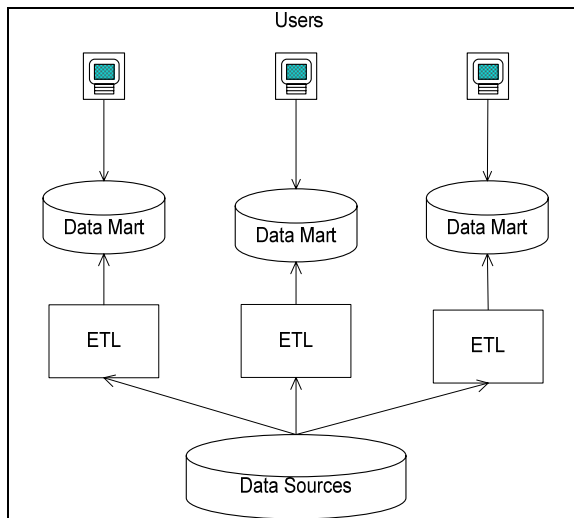


Figure 4: Independent data marts

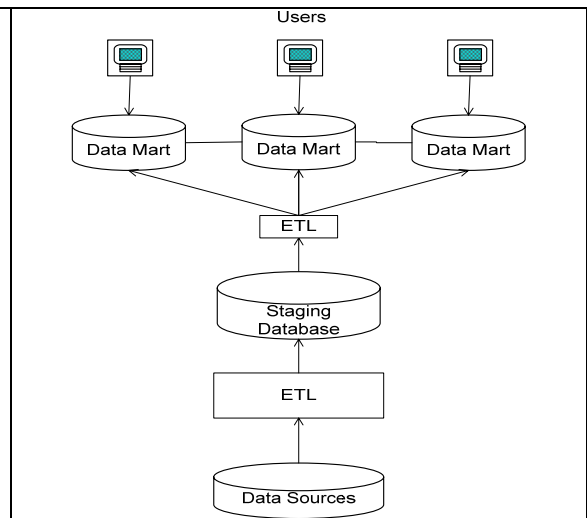


Figure 5: Data mart bus architecture with conformed dimensions

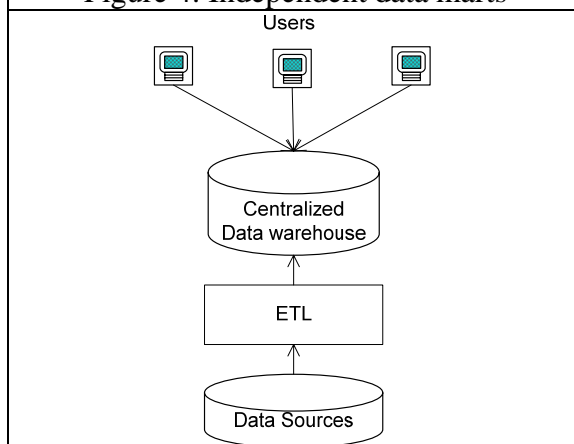


Figure 6: Centralized DW architecture with no data marts

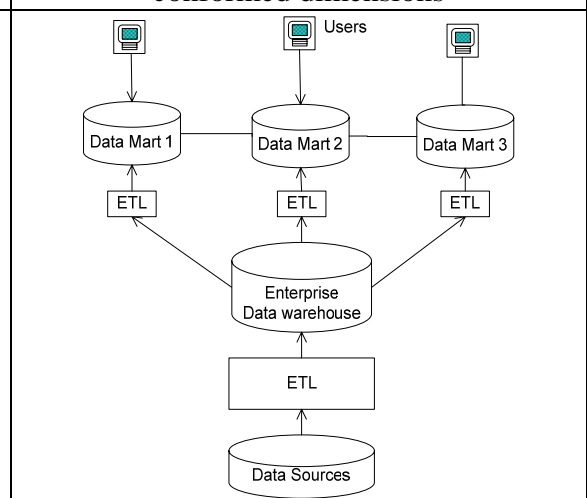
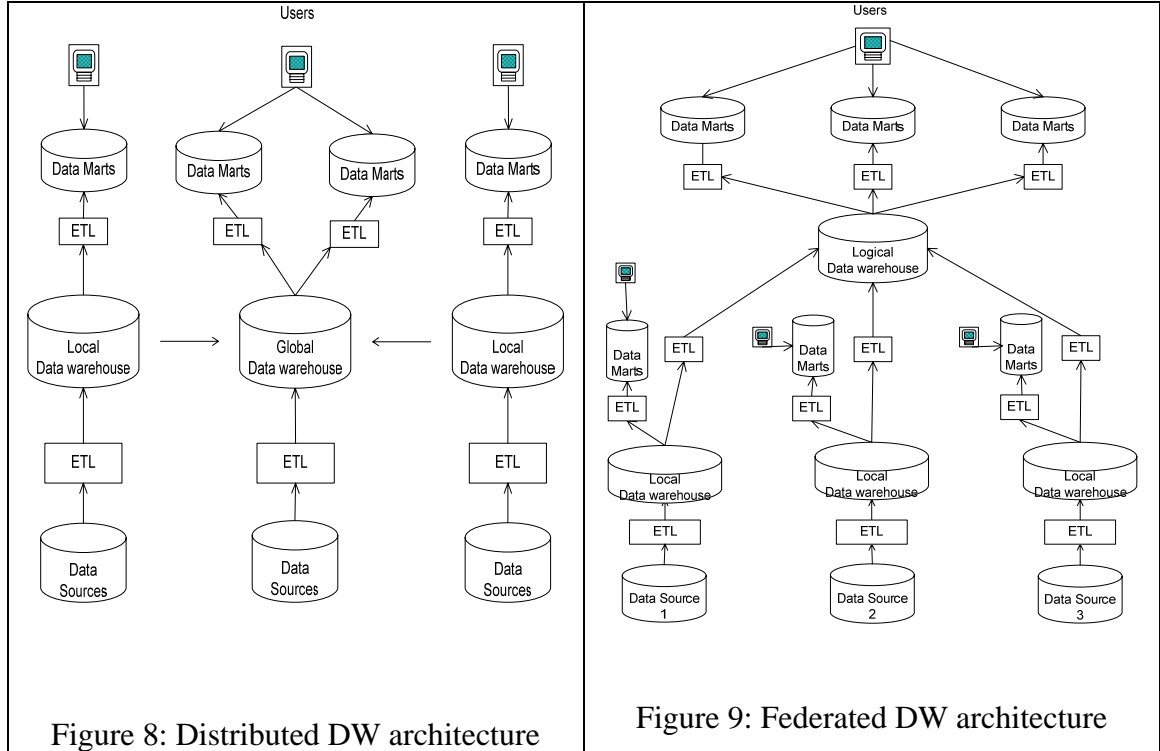


Figure 7: Hub-and-Spoke Architecture



Independent Data Marts Architecture

In this architecture, multiple data marts are created independently of each other. The data marts do not use conformed dimensions and measures. Thus, there is no unified view of enterprise data in this architecture. As the number of data marts grows, maintenance of consistency among data marts are difficult. In the long run, this architecture is likely to produce “silos of data marts.”

Data Mart Bus Architecture with Conformed Dimensions

In this architecture, instead of creating a single enterprise level DW, multiple dimensional data marts are created that are linked with conformed dimensions and measures to maintain consistency among the data marts [6, 7]. Here, an enterprise DW is a union of all the data marts together with their conformed dimensions. The use of the conformed dimensions and measures allows users to query all data marts together. Data marts contain either atomic data or summary data. The strength of the architecture is that data marts can be delivered quickly, and multiple data marts can be delivered incrementally. The potential weaknesses are that it does not create a single physical repository of integrated data and some data may be redundantly stored in multiple data marts.

Centralized Data Warehouse Architecture

In this architecture, a single enterprise level DW is created for the entire organization without any dependent data marts. The warehouse contains detailed data for all the analytic needs of the organization. Users and applications directly access the DW for analysis.

Hub-and-Spoke Architecture (Corporate Information Factory)

In this architecture, a single enterprise DW, called the hub, is created with a set of dimensional data marts, called spokes, that are dependent on the enterprise DW. The warehouse provides a single version of truth for the enterprise, and each data mart addresses the analytic needs of a business unit. This architecture is also called the corporate information factory or the enterprise DW architecture [3]. The warehouse contains data at the atomic level, and the data marts usually contain either atomic data, lightly summarized data, or both, all fed from the warehouse. The enterprise warehouse in this architecture is usually normalized for flexibility and scalability, while the data marts are structured in star schemas for performance. This top-down development methodology provides a centralized integrated repository of the enterprise data and tends to be robust against business changes. The primary weakness of this architecture is that it requires significant up-front costs and time for developing the warehouse due to its scope and scale.

Distributed Data Warehouse Architecture

A distributed DW architecture consists of several local DWs and a global DW [3]. Here, local DWs have mutually exclusive data and are autonomous. Each local warehouse has its own ETL logic and processes its own analysis queries for a business division. The global warehouse may store corporate-wide data at the enterprise level. Thus, either corporate-level data analysis at the enterprise level or global data analyses that require data from several local DWs will be done at the global DW. For example, a financial analysis covering all the business divisions will be done at the global DW. Depending on the level of data and query flows, there could be several variations in this architecture [3]. This architecture supports multiple, geographically distributed business divisions. The architecture is especially beneficial when local DWs run on multiple vendors.

Federated Data Warehouse Architecture

A federated DW architecture is a variation of a distributed DW architecture, where the global DW serves as a logical DW for all local DWs. The logical DW provides users with a single centralized DW image of the enterprise. This architecture is a practical solution when an enterprise acquires other companies that have their own DWs, which become local DWs. The primary advantage of this architecture is that existing environments of local DWs can be kept as they are without physically restructuring them into the global DW. This architecture may suffer from complexity and performance when applications require frequent distributed joins and other distributed operations.

Virtual Data Warehouses Architecture

In a virtual DW architecture, there is no physical DW or any data mart. In this architecture, a DW structure is defined by a set of materialized views over OLTP systems. End-users directly access the data through the materialized views. The advantages of this approach are that it is easy to build and the additional storage requirement is minimal. This approach, however, has many disadvantages in that it does not allow any historical data; it does not contain a centralized metadata repository; it does not create cleansed standard data items across source systems; and it could severely affect the performance of the OLTP system.

KEY APPLICATIONS

Numerous business applications of data warehousing technologies to different domains are found in [1, 6]. Design and development of clickstream data marts is covered in [5].

Applications of data warehousing technologies to customer relationship management (CRM) are covered in [2, 11]. Extension of data warehousing technologies to spatial and temporal applications is covered in [8].

URL to CODE

Two major international forums that focus on data warehousing and OLAP research are International Conferences on Data Warehousing and Knowledge Discovery (DaWaK) and ACM International Workshop on Data Warehousing and OLAP (DOLAP). DaWaK has been held since 1999, and DOLAP has been held since 1998. DOLAP papers are found at <http://www.cis.drexel.edu/faculty/song/dolap.htm>. A collection of articles on industrial DW experience and design tips by Kimball is listed in <http://www.ralphkimball.com/>, and the one by Inmon is listed in www.inmoncif.com.

CROSS REFERENCES

Active and Real-time Data Warehousing, Cube, Data mart, Data mining, Data warehouse, Data warehouse life-cycle and design, Data warehouse maintenance, evolution and versioning, Data warehouse metadata, Data warehouse security, Dimension, Extraction, transformation and loading, Materialized views, Multidimensional modeling, On-line analytical processing, Optimization and Tuning in data warehouses, View maintenance

RECOMMENDED READINGS

- [1] Adamson, C., Venerable, M. (1998): Data Warehouse Design Solutions. Wiley.
- [2] Cunningham, C., Song, I.-Y., Chen, P.P. (2006): Data Warehouse Design for Customer Relationship Management. Journal of Database Management, 17(2): 62-84, 2006.
- [3] Inmon, W.H. (2002): Building the Data Warehouse, Third edition, Wiley.
- [4] Jones, M.E, Song, I.-Y. (2008): Dimensional Modeling: Identification, Classification, and Evaluation of Patterns. Decision Support Systems, 45(1): 59-76.
- [5] Kimball, R., Merz, R. (2000): The Data Warehouse Toolkit: Building the Web-Enabled Data Warehouse. Wiley.
- [6] Kimball, R., Ross, M. (2002): The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. Second edition. Wiley.
- [7] Kimball, R., Ross, M., Thorntwaite, W., Munday, J., Becker, B. (2008): The Data Warehouse Lifecycle Toolkit, Second edition. Wiley.
- [8] Malinowski, E., Zimanyi, E. (2008): Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications. Springer.
- [9] Poole, J., Chang, D., Tolbert, D., Mellor, D. (2002): Common Warehouse Metamodel: An Introduction to the Standard for Data Warehouse Integration. Wiley.
- [10] Sen, A., Sinha, At.P. (2005): A Comparison of Data Warehousing Methodologies. CACM, 48(3): 79-84.
- [11] Todman, C. (2000): Designing a Data Warehouse Supporting Customer Relationship Management. Prentice Hall.
- [12] Watson, H.J., Ariyachandra, T. (2005): Data Warehouse Architectures: Factors in the Selection, Decision, and the Success of the Architectures". From http://www.terry.uga.edu/~hwatson/DW_Architecture_Report.pdf.