IDEA GROUP PUBLISHING

ITJ2223

1331 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

A Metadata Oriented Architecture for Building Datawarehouse[1]

HEESEOK LEE, TAEHUN KIM, AND JONGHO KIM
Korea Advanced Institute of Science and Technology, Korea

Data warehouse is an intelligent store of data that can aggregate vast amounts of information. A metadata is critical for implementing data warehouse. Therefore, integrating data warehouse with its metadata offers a new opportunity to create a more adaptive information system. This paper proposes a metadata-oriented data warehouse architecture that consists of seven components: legacy system, extracting software, operational data store, data warehouse, data mart, application, and metadata. A taxonomy for dataflow and metaflow is proposed for better understanding of the architecture. In addition, a metadata schema is built within the framework of the seven components. The architecture with its metadata component is applied to a real-life data warehouse for a large medical center in order to illustrate its practical usefulness.

Data warehouse (DW) has progressed quickly, and its related technologies are constantly being introduced. However, academic research for DW is rare (McFadden, 1996). Information technology strategists have attempted to understand "what" issues of DW and only now start to consider "how" issues (Atre & Storer, 1995). It is important to take an academic look, rather than a vendor-oriented glance, at DW.

Inmon and Hackathorn (1994), the "father" of DW, defined it as "*an integrated, time-variant, subject-oriented*, and *nonvolatile* collection of data in support of management's decision making process", but DW is merely a new name for an old idea-that of making an enterprise's data more accessible (Hoven, 1997). Thus, DW requires a fundamental architecture for enterprise information processing. An operational database which continuously produces operational data (Chaudhuri & Dayal, 1997) is based on on-line transaction processing (OLTP) applications, whereas DW is based on on-line analytical processing (OLAP) applications. DW continuously produces analytical information for business users.

Building DW requires two important development issues: (i) DW for the decision making of business users and (ii) metadata within it. Typically, metadata is defined as data about data. For example, the definitions of tables, columns, databases, views, and other objects are all metadata (Gill and Rao, 1997; Tozer, 2000). It is like a roadmap. For example, in the view of a library borrower, a card catalog (i.e., metadata) points to the contents and location of a book (i.e., data) (Tannenbaum, 1994). This paper defines metadata as information about DW, i.e., information about queries, reports, transformations, tables, columns, and users.

Most DW development methodologies have not considered metadata development; it is necessary to adopt an integrated methodology which develops a DW and its metadata simultaneously. Metadata is a key to success of data warehousing system. That is, metadata is crucial documentation for a data warehousing system where users should be empowered to meet their own information needs (Bischoff & Alexander, 1997); users need to know what data exists, what it represents, where it is located, and how to access it. Furthermore, metadata is used for extracting data and managing DW. However, metadata has failed because its management has been segregated from the DW development process (Denzer & Guttler, 1996; Kutsche & Sünbül, 1999). Metadata must be integrated with data warehousing systems. Without metadata, the decision support of DW is under the control of technical users. As DW evolves, extracting data from OLTP systems to DW becomes more complex. If metadata is integrated with DW, the extraction can be automatic. When metadata is separated from the DW development process, many tools can have a set of mutually exclusive metadata that describes the same data (Anahory & Murray, 1997).

For this integration, this paper proposes taxonomies for dataflow and metaflow. Furthermore, an architecture for DW is developed on the basis of these taxonomies. The architecture consists of seven components: legacy, extracting, operational data store (ODS), DW, data mart (DM), application, and metadata. The emphasis is on the metadata component, which consists of technical metadata and business metadata. A generic metadata schema is built for DW. Because the data warehousing system is built mainly for business users, the business metadata is further explored. The architecture, with its metadata component, is applied to implementing a real-life DW in order to illustrate its practical usefulness.

DATA WAREHOUSE: SYSTEM, TAXONOMY, AND ARCHITECTURE

Data Warehousing System

Users of data warehousing systems are classified into business users and technical users. Business users include executive users, casual users, business analysts, and power users (Poe, 1997). Technical users include system administrators, application developers, operators, technical supporters, and designers. The framework for data warehousing system may be derived from an input-process-database-output system by Orr (1998). The input corresponds to the operational database. The process corresponds to the extracting software. The database includes informational database which consists of DW, ODS (Inmon et al., 1997), and DM (Hackney, 1997; Hoven, 1998). The output utilizes the database, i.e., application software. A metastore, a database containing metadata, controls and manages the other parts.

The term "data warehousing" is used to emphasize the dynamic characteristics of DW (Hackathorn, 1995). The dynamics of DW are expressed by the following five flows.

Taxonomies for Dataflow and Metaflow

Taxonomies for dataflow and metaflow are important because they can describe all of the processes of using data warehousing system (Lee et al., 1997). Hackathorn (1995) originally investigated five flows on which taxonomies are based: inflow, upflow, downflow, outflow, and metaflow. This paper adopts this classification (Table 1).

Inflow feeds data from legacy systems and other external sources into the informational database or users' application. In this process, data from the operational system may be transformed. Upon entering the informational database, data

Table 1: Dataflow and Metaflow

Flows			Explanation
Data flow	Inflow		Operational databases (including data from external sources) are cleansed and proceed to informational database.
	Ownflow	Upflow	Highly detailed data are aggregated and summarized.
		Downflow	Old data is archived or deleted according to a purge criteria.
	Outflow		End-users get data from informational databases by running canned or ad-hoc queries.
Metaflow			Processes that move metadata, which come from several flows.

become highly detailed. Upflow aggregates and summarizes the highly detailed data. Downflow purges or archives data into storage media such as magnetic tape. It helps maintain the vitality of informational database. Outflow implies that users utilize data from the informational database by a simple or an advanced tool with functions ranging from basic management reporting to complex, drill-down, and analytical processing.

Metadata are related with the other four flows. Metaflow keeps metadata up to date. The metaflow can depend on a particular DBMS (Database Management System), and its content can vary depending on the purpose of its usage.

In this paper, we propose taxonomies for dataflow and metaflow. The taxonomy for dataflow has three dimensions: inflow, ownflow, and outflow. Metaflow can be classified according to three criteria: situation, distribution, and usage.

Figure 1 depicts the taxonomy for dataflow. The dimension called inflow implies how to access legacy systems. Inflow can be one of three types (Widom, 1995). First, end-

Figure 1: Taxonomy for Dataflow