통상정보연구 제7권 제1호 (2005년 3월 25일)
논문투고일자 : 2005. 02. 17
게재확정일자 : 2005. 03. 21

Implementing Data warehouse Methodology Architecture: From Metadata Perspective

Sang-youl Kim*

Graduate School of Internatinal Studies, Pusan National University, Assistance Professor

Tae-hun Kim

Department of Digital Business, Kyungsung University, Assistance Professor

Contents

Ⅰ. Introduction
Ⅱ. Methodology Comparison and Past Studies
Ⅲ. Proposed Methodology
Ⅳ. Validation of proposed methodology
References

Abstract

Recently, many enterprises have attempted to construct data warehousing systems for decision-support. Data warehouse is an intelligent store of data that can aggregate vast amounts of information. Building DW requires two important development issues:(i) DW for the decision making of business users and (ii) metadata within it. Most DW development methodologies have not considered metadata development; it is necessary to adopt a DW development methodology which develops a DW and its metadata simultaneously. Metadata is a key to success of data warehousing system and is critical for implementing DW. That is, metadata is crucial documentation for a data warehousing system where users should be empowered to meet their own information needs; 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. Metadata must be integrated with data warehousing systems. Without metadata, the decision support of DW is under the control of technical users. Therefore, integrating data warehouse with its metadata offers a new opportunity to create a more adaptive information system.

Therefore, this paper proposes a DW development methodology from a metadata perspective. The proposed methodology consists of five phases: preparatory, requirement analysis, data warehouse (informational database) development, metastore development, and maintenance. To demonstrate the practical usefulness of the methodology, one case is illustrated

Key Words : Data Warehouse, Metadata, Development Methodology, Architecture

* First Author

Ⅰ. Introduction

Data warehouse (DW) concept has progressed quickly, and new 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 and 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" (p. 2), 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 and 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. The term OLAP has been invented recently to represent the alternative for OLTP (Devlin, 1997).

Building DW requires two important development issues: (i) DW for the decision making of business end-users and (ii) metadata within it. Metadata is popularly defined as data about data. For example, the definitions of tables, columns, databases, views, and other objects are all metadata (Tozer, 1999; Gill and Rao, 1996). It is like a roadmap to data. 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; McClure, 1992; Froeschl, 1997; Bouziane, 1991; Timpf et al., 1996). This dissertation uses the term metadata as information about data warehousing system, e.g., information about queries, reports, transformations, tables, columns, and users. Because DW evolves, we can extend the concepts of metadata by using the definition.

Most DW development methodologies have not considered metadata development. Metadata is generally considered as software in legacy methodologies; there is an increasing necessity to adopt a DW development methodology which develops a DW and its metadata simultaneously. Metadata is a key to success of data warehousing system (Griffin, 1997). That is, metadata is crucial documentation for a data warehousing system where users should be empowered to meet their own information needs (Bischoff and 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. The metadata can provide an enterprise with the standardized data element, when the processes use the element. However, metadata has failed because its management has been segregated from the DW development process (Denzer and Guttler, 1996; Kutsche and Sunbul, 1998). Metadata must be integrated with data warehousing systems. Without metadata, end-users may depend on their own knowledge or ask technical users for information about data. That is, 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. In the unintegrated case, if data source is changed or added, it means that DW is manually evolved and developed. In general, many tools use metadata. 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 and Murray, 1997).

The primary objective of this paper is to propose a new methodology for developing metadata within a data warehousing system. For the objectives, this paper proposes a metadata-based development methodology for building a data warehousing system. Most of the existing methodologies have not considered metadata development; to adopt a metadata tool or nothing. Furthermore, they have not regarded a data warehousing system as processing evolutionarily. Based on the metadata model, an iterative development methodology is proposed from the standpoint of an evolution of a data warehousing system.

Ⅱ. Methodology Comparison and Past Studies

1. Methodology Comparison

Table 1, in which we compared the methodology with others' methodologies, shows some characteristics and benefits of our methodology. We have not only included the technical and business metadata but also adopted both the top-down and bottom-up approach. In addition, we compare the architecture of the current study with those of the previous studies and other systems in further detail (See Table 1).

Few architectures of researches consider ODS (Operational Data Store) components and the development of metadata. Also, most consider DW,-DM relationship from a top-down approach, in which DW provides data to DMs. Commercial systems such as NCR and Oracle are dependent on products they have developed and have an exclusive metadata product. They, that is, do not have an architecture, but an infrastructure. Most other methodologies do not take into consideration of metadata development with some expectations.

Because architecture is akin to skeletal structure, the other architectures appear similar to the proposed ones, considering the results. Although the proposed architecture appears similar to other architectures that involve metadata, many differences are found in the paper: the vendors rarely differentiate between architectures and infrastructures. The architectures of vendors are tool-oriented while those of non-vendors focus on the solutions that are not specific to any tool. The architecture provides a foundation of building a DW by the use of the metadata component, not buying it. Also, we consider ODS, DM, and top-down & bottom-up approach. The architecture is based on the taxonomies. In addition, this study does not include commercial articles and papers about architecture so far as possible, though there's been a little research recently in this field.

〈Table 1〉 Comparison of Methodology

Research / Items	Inmon (1993)	Inmon, Imhoff & Battas (1996)	Noaman & Barker (1997)	Murtaza (1998)	Oracle (Gray & Watson (1998))	NCR (Gray & Watson (1998))	Jarke et al. (1999)	This Research
Modeling Method	ER	Star-shaped	N/A	N/A	Star-shaped	N/A	N/A	Star-shaped
Metadata Development	N/A	N/A	N/A	N/A	N/A	N/A	Meta Database	Metastore
Major Phases	10 Steps	6 Sections	N/A	5 Stages	N/A	3 Phases	N/A	7 Phases
Development Method	Iterative Waterfall	Waterfall	N/A	N/A	N/A	Waterfall	N/A	Iterative Waterfall
Method Name	Data Driven Development Method	ODS Development Method	N/A	5-stage Framework	N/A	NCR DW Method	N/A	DW Development Methodology
Considering Other Informational DB	DM Development	N/A	N/A	N/A	N/A	N/A	N/A	DW, ODS, and DM Development
Comments	Architectural Development DM-CLDS	ODS Development Detailed Methodology Deliveries	Distributed DW Architecture	Architecture -driven Approach	Package Product Solution DM and Applications	Numerous Alliance Partners	Quality Architecture	Taxonomy Architecture Metadata Development

Existing methodologies rarely differentiate architecture and infrastructure in the case of vendors' methodologies. Their methodologies are tool-oriented while non-vendors' focus on the solutions that are not specific to some tools. Also, there are less research on the methodologies for ODS and DM development. Inmon (1993) mainly focuses on DW development which has 10 steps using ER for modeling. He also follows top-down approach for DM development. Poe (1997) focuses on DW development with 10 steps using star-shaped schema for modeling. Hackney (1997) focuses on DM development using star-shaped schema for modeling. He adopts spiral model which can accommodate both top-down and bottom-up approach. NCR has tool-oriented solution adopting waterfall model.

2. Past Studies

About the methodology aspect, Inmon (1993) proposed a data-driven methodology. The data-driven methodology is presented in three parts: METH1, METH2, and METH3. The first part of the methodology is for operational systems and processing. METH2 is for the DW. The third part of the methodology describes the usage of the DW. We focus at METH2. This methodology does not take an application approach to the development of systems. Whereas operational development is shaped around a development life cycle that begins with requirements and ends with code, this methodology begins with data and ends with requirements. The methodology is comprised of ten steps: data model analysis, breadbox analysis, technical assessment, technical environment preparation, subject area analysis, DW design, source system analysis, specification, programming, and population. This methodology is suitable for a data warehouse, not for a data warehousing system.

Poe (1997) proposed a decision support life cycle (DSLC). The phases of DSLC is comprised of ten phases: planning, gathering data requirements and modeling, physical database design and development, data mapping and transformation, data extraction and load, automating the data management process, application development, data validation and testing, training, and rollout. But, the methodology is a kind of waterfall model. Though the goals and the data structures of on-line transaction-based and decision support systems differ, DSLC does not deviate from the classical system development life cycle (SDLC). DSLC is accomplished for only one process. However, the SDLC method is ill-suited to decision-oriented applications (Laudon and Laudon, 1994); it is still used for building large OLTP systems. Decision-making can be rather unstructured and flexible; requirements constantly change and decisions may have no well-defined models or procedures. Decision-makers often cannot specify their information needs in advance, and sometimes must experiment with concrete systems to clarify the kinds of decisions they wish to make. This high level of uncertainty cannot be easily accommodated by SDLC.

Thus, prototyping is particularly true of decision-oriented applications, where requirements tend to be very vague. Yet it is inappropriate for large complex systems (Laudon and Laudon, 1994). Instead, prototyping may be appropriate for DM system; thus, in data warehousing system, we have need of composite method for the iteration of user evaluation and large complex systems. The iterative SDLC will have met with mixed success. It has the iterative approach with characteristics of SDLC.

Murtaza (1998) proposed a five-stage framework: business requirements, data sourcing, target architecture, access tool selection, and data warehouse administration. Because this framework is an architecture-driven approach that subsequently involves selection of add-on data warehouse tools and meeting ongoing administration requirements, this framework does not suggest development processes and method in great detail.

Jarke et al. (1999) proposed an implementation of metadata database. The goal of their work is to enrich metadata management in data warehouses such that it can serve as meaningful basis of systematic quality analysis and quality-driven design.

Noaman and Barker (1997) explained a centralized data warehouse architecture and discussed several models for distributed data warehouse architectures and an alternative model. Thus, a methodology related to this research is not covered.

Hackney (1997) proposed a spiral methodology for building a data mart. The methodology has eleven discrete steps: business driver, design, build, test, train, pilot, deploy, use, monitor, maintain, and identify additional business driver. He also explained the differences and commonalties between subset (top-down) and incremental (bottom up) data marts.

Inmon et al. (1996) proposed an ODS development methodology which is divided into six sections: project management, pre-requisite, process design, data design, system testing, and end-user environment. Though they presented steps of developing a metadata repository and access, we do not consider a metadata development because the steps are not based on a metadata model. The data design section is used to create the ODS database. The section includes the creation of the ERD, logical and physical data models, mapping of the source system of record, and population of the ODS database.

Gray and Watson (1998) investigated methodologies of commercial systems: NCR and Oracle. NCR has a NCR data warehouse development which is comprised of three phases: planning, design and implementation, and usage, support, and enhancement. This methodology also provides a logical data model design process. Activities include the confirmation of requirements, creation of a project plan, and the generation of the logical data model showing relationships and attributes. Also, NCR has maintained numerous alliances with industry leaders, e.g., Microsoft and Sun Microsystems. The information of an Oracle's development methodology is not available in Gray and Watson's work (1998). There are mainly contents of data mart products and application products.

Ⅲ. Proposed Methodology

Figure 1 shows a DW development methodology, used in the development of a data warehousing system.

As laid out in the Figure, this methodology is composed of seven phases: preparatory, requirement analysis, DW development, ODS development, DM development, maintenance, and metastore development.

The preparatory phase has two stages: strategy planning stage and DW architecture and infrastructure stage. In the first stage, DW subject candidate is found out and its contents are evaluated. Three output tables are produced in this stage (Subramanian et al., 1997); (i) a functional profile table, which describes the mission and functions of the department and its subdivisions; (ii) an information types table, which describes the types of information used or generated by the department; and (iii) an information usage matrix table, which describes the associated computing platform or storage medium and the functions supported by the information, its transmission to or from other departments or external organizations, and the currency of the various types of information. The ultimate purpose of this stage is to find candidates for data warehousing subjects.

In the second, DW architecture and infrastructure stage, architecture and infrastructure is determined in accordance with the subject area drawn out in the preparatory phase. In other words, the framework, outline, and scope of a project are defined. Three results are produced in this stage: (i) taxonomy for dataflow; (ii) a taxonomy for metaflow; and (iii) a DWAI table that summarizes the technical information regarding each and chosen component. The preparatory phase is the starting point of an evolutionary and iterative process, and its results are input factors in the next, requirement analysis phase. The preparatory phase thus provides a comprehensive perspective on DW projects.

Requirements are classified into two categories: operational and informational (Inmon and Hackathorn, 1994); the latter is further divided into DW or DM requirements. Therefore, requirement analysis consists of three elements: ODS, DW, and DM requirements. Those requirements are input factors in the ODS, DW, or DM development phase. The objective of requirement analysis phase is to understand the user requirements and formulate them. Three results are produced: (i) a dimensional business model displaying metrics, dimensions, and relationships, presented to business users for verification, (ii) an attribute hierarchy table, and (iii) a fact information table.

The dimensional business model could be a graphical description of the simple star schema, in which case the logical and physical models will, with the exception of physical table partitioning, be identical (Poe, 1997). A traditional conceptual model, such as ER could be applied to DW development instead (Silverston et al., 1997; Golfareli et al., 1998; Poe, 1997); but the dimensional business model is better than ERD (Entity Relationship Diagram) (Kimball, 1996). The attribute hierarchy shows a parent-child relation of attributes in one dimension. It also contains information on stem and relative attributes. The former is an attribute set capable of doing a roll-up and drill-down analysis. The latter is attributes that are not viewed as hierarchy structure but viewed as the same level with a stem attribute (Thomsen, 1997). The fact information table describes additive and semi-additive variables.

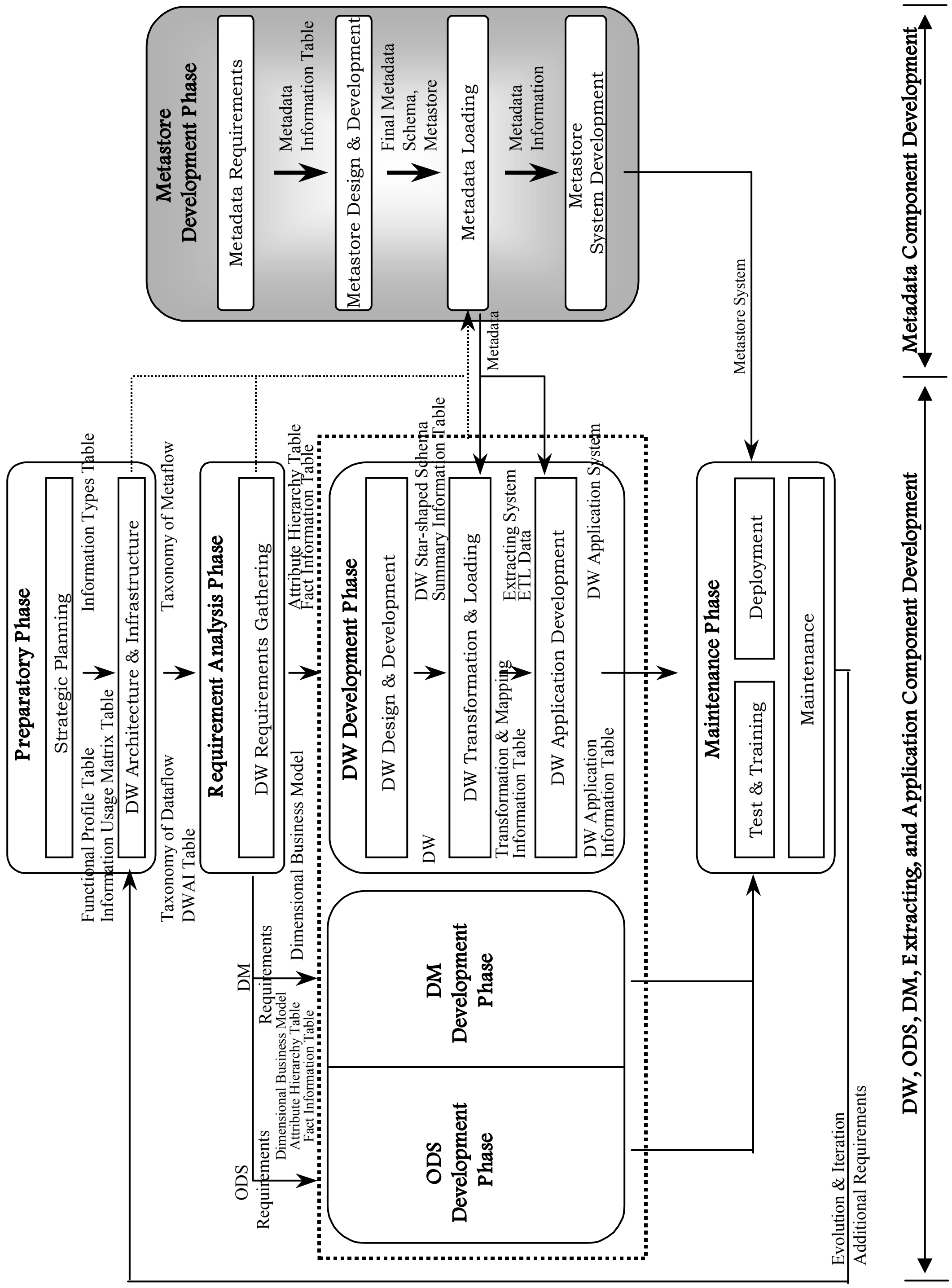
In the DW development phase, the dimensional business model is transformed into a logical DW model, DW and a physical model containing aggregations and partitions are implemented, and applications are developed. The phase has three stages: (i) DW design and development stage, (ii) DW transformation and loading stage, and (iii) DW application development stage.

The DW design and development stage produces three results: (i) summary information table that contains information on derived variables aggregations, (ii) star-shaped schema such as star, snowflake or mixed schema, and (iii) DW that is physically implemented. The DW transformation and loading stage is a process during which operational data is extracted, filtered, validated, merged, aggregated, and finally loaded to DW. The stage produces three results: (i) a transformation & mapping information table showing the relation between source data and DW data, (ii) extracting system, and DW loaded with data through ETL (extracting, transforming, and loading) process. The transformation & mapping information table contains criteria, cardinality information between operational database attribute and DW attribute, source information, etc. Criteria are five kinds: data conditioning, integration, stasis, simplification, and amplification (Moriarty and Mandracchia, 1996). In this stage, extracting system can be done either by purchasing a product or by individual development based on metadata.

The DW application development stage based on the developed DW has two results: (i) DW application information table that contains information on canned queries and reports, and (ii) DW application system implemented by the use of metadata or DW-exclusive application software for commercial purpose.

The ODS and DM development phase is similar to DW development, yet it has different domain and purpose (Chaudhuri and Dayal, 1997); similar applications are produced in this phase.

Information from the previous 5 phases is delivered to the metastore development phase. Metastore development phase is composed of four stages: (i) metadata requirements, (ii) metastore design and development, (iii) metadata loading, and (iv) metastore system development. In the metadata requirements stage, requirements are understood through interviews with business users and technicians. The result is metadata information table. The domain-specific and additive metadata are extracted through the interviews and displayed on the metadata information table. In the metastore design and development stage, logical modeling of metadata and implementation of metastore are done. This stage has two results: (i) final metadata schema produced by adding domain-specific metadata to core metadata schema, and (ii) metastore. Metadata loading is a process during which metadata is loaded to metastore. In Figure 1, the dotted-line arrow shows that information produced from each previous stage becomes an input factor in this stage. The metastore system development stage refers to the system that searches and manages the metadata at metastore.



〈Figure 1〉 Methodology Architecture

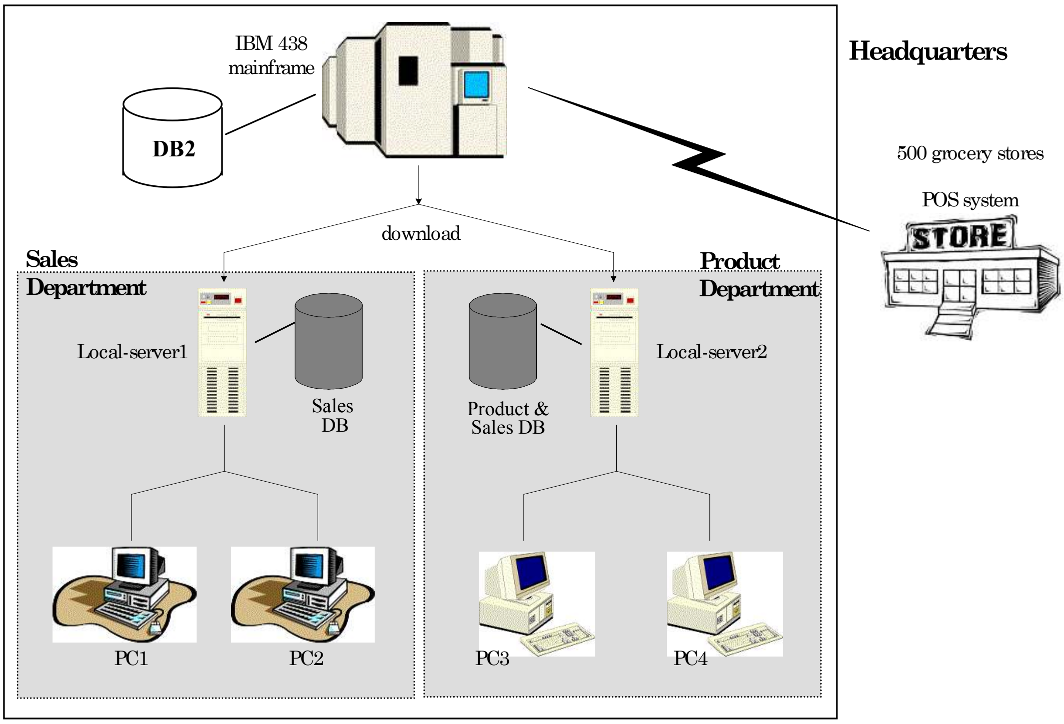
Completion of all the development stages is followed by the maintenance phase. The phase consists of three stages: (i) test and training, including testing of access tools and application systems, validating data, and proactively supporting users; (ii) deployment (Poe, 1997); and (iii) maintenance (Gill and Rao, 1996). During this phase, whether the data warehousing system should evolve or not is considered. New-incoming requirements such as new DM, or new source become the factors in changing data warehouse architecture (DWA), and start to iterate every phase. In this evolutionary process, data warehousing system goes through a gradual progress in regard to DWA and user satisfaction. Prior phases are explained mostly in terms of its relation with DW-oriented development. For instance, we can consider DM development from the perspective of bottom-up approach. After the completion of requirement analysis phase, DM development phase illustrated in Figure 1 can be pursued. Following the maintenance phase, DWAI phase, requirement analysis phase, and DW development phase can be subsequently completed.

Separation of business metadata and technical metadata is not clear when it comes to the use of metadata. Further, business metadata and technical metadata may be overlapped with user requirements. This stage is a process where business metadata and technical metadata are achieved, and the focus is given on the understanding of business metadata. In Figure 1, the left part of metastore development phase is the process where mostly technical metadata is achieved.

Ⅳ. Validation of proposed methodology

The example was chosen from the case of a grocery store in Kimball's book (1996). The database in the example is a hypothetical one. We will imagine we work in the headquarters of a large grocery chain. Our business has 500 large grocery stores spread over a three-state area. Each of the stores is a typical modern supermarket with a full complement of departments including grocery, frozen foods, dairy meat, produce, bakery, floral, hard goods, liquor, and drugs. Each has roughly 60,000 individual products on its shelves. About 40,000 of the individual products come from outside manufacturers and have bar codes imprinted on the product package. The grocery stores scan the bar codes directly into the point of sale (POS) system. The most significant management decisions that can be made in real time have to do with pricing and promotions. Both Sales department and Product department at headquarters spend a great deal of time tinkering with pricing and running promotions. Promotions in the grocery store include temporary price reductions, advertisements in newspapers and newspaper inserts,

displays in the grocery store including shelf displays and end-aisle displays, and coupons.



<Figure 2> System Configuration

Figure 2 shows the current system configuration in operation. Through the POS system, information is collected into the IBM 438 mainframe at the headquarters. Sales department and Product department download data from the mainframe on a daily basis. Each of them has one local server with which they manage the database belonging to their department. The database at local server 1 stores information on sales, corporation, store, and promotion. The database at local server 2 stores information on sales, product, and vendor. The Sales department has two personal computers (PCs) for analysis. PC1 is used by an end-user for analyzing the promotional information on sales. PC2 is used by an end-user for analyzing the corporate and store information on sales. Product department has two PCs for analysis. PC3 is used for analyzing the product sales information in regard to the sales and products. PC4 is used for analyzing the vendor information on sales and products.

Because of inconsistent data, those in charge of departments are not capable of analyzing sales and product information from various perspectives. Information from each department should be integrated(stored together), so that a new environment where sales and products information is analyzed from a single, overall perspective can be created. The newly introduced data warehousing system will provide such a new environment to the company.

1. Preparatory Phase

The scope of data warehousing system was limited to the Sales department and Product department. By understanding the current situation of these two departments, a functional profile in Table 2 is produced. The mission and function of Sales department include Sales Analysis Program which analyzes sales of a grocery store, Corporation Program which manages goods delivered to corporation customers, and Store and Promotion Program which develops promotional events and items. The two departments also collect and manage information on all of the 500 grocery stores.

<Table 2> Functional Profile

Department Name: Sales Department	
Area	Sales Analysis and Responsibility for Branch Office Supmter Lee, Sales Manager 325-4539
Mission	To administer all of the following store programs: 1) Sales Analysis Program 2) Corporation Program 3) Store and Promotion Program
Functions	1) analyze the sales data 2) manage goods delivered to the corporation customer 3) develop promotional events and items, and administer distributed stores

Department Name: Product Department	
Area	Product Development and Management Meyer Poll, Product Manager 325-4542
Mission	To administer all of the following product programs: 1) Product Management Program 2) Product Vendor Program 3) Product Analysis Program
Functions	1) administer product about it and develop the new product 2) manage the product vendors and gather information about vendors 3) analyze the product sales data

1) Data Warehouse Architecture and Infrastructure

The project scope focuses on sales, product, promotion, and store information, which put bigger importance on frozen data than on current ones. Grain is based on the daily granularity shows.

On the basis of the taxonomy for dataflow, this case adopted in-advance type, single, and access type. Also, based on the taxonomy for metaflow, stand-alone type, business type, and centralized type were adopted as metaflow. Types chosen in each dimension influence next phases of this case. For instance, the access type of outflow dimension is proceeded by choosing a simple report tool in this case.

Table 3 shows DWAI. Only DW component is chosen in the informational database. The infrastructure is subject to the chosen architectural component. After an architectural component is selected, relevant infrastructure is purchased or developed. Application component in the architecture has two application programs. The one is a copy of Star Tracker Solo software, and the other is an access application software developed by using Microsoft Visual Basic 5.0.

<Table 3> DWAI

Architecture	Infrastructure
Legacy Component (4)	DB2 Database (Mainframe) Sales Database (Local-server 1) Product and Sales Database (Local-server 2) External Database (Bloomberg News)
Extracting Component (1)	InfoPump ver 2.1 (Purchase)
DW Component (1)	NT Server 4.0 Platform (Purchase)
	MS SQL Server 6.5 (Purchase)
Metadata Component (1)	Metastore System (Development)
Application Component (2)	Reporting Application By MS VB 5.0 (Purchase)
	Star Tracker Solo Ver 2.1.0. (Purchase)
Other Factors	PowerSoft WarehouseArchitect 6.0 (Purchase)

2. Requirement Analysis Phase

Through interviews with business end-users about DW candidates, the information they need can be clarified. Based on this, information items in the existing system are compared and analyzed to produce attribute hierarchy table and fact information table. Table 4 and Table 5 show the information thus produced. We assume the logical model is identical with physical model in this case, dimensional business model will be similar to star-shaped schema. That is the reason dimensional business model is omitted in this section. In the attribute hierarchy table, elements of stem have one-to many cardinality from top-down. In the time hierarchy table, the relative of date is Day-of-Week element. In the sales fact information table, Dollar-Sales, Unit-Sales, and Dollar-Cost were selected as additive variables. As a semi-additive variable, Customer-Count variable is selected. Such information is revealed in more detail and utilized in the design stage.

<Table 4> Attribute Hierarchy

Relation \ Dimension	Time		Product	
	Stem	Relative	Stem	Relative
Parent (Drill-down ↓, Roll-up ↑)	Year	–	Department	–
	Quarter	–	Category	–
	Month	–	Brand	–
Child	Date	Day_of_Week	Product	Weight, Package_Size

<Table 5> Fact Information

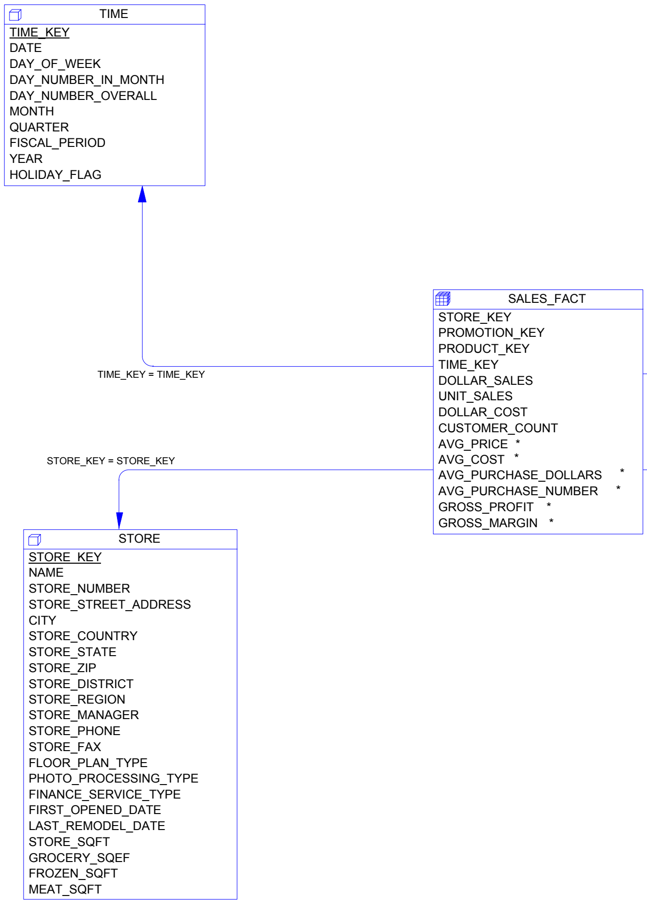
Variable \ Table	Sales
Additive	Dollar_Sales Unit_Sales Dollar_Cost
Semi-additive	Customer_Count

3. Data Warehouse Development Phase

Figure 3 shows a star-shaped schema in which logical design is identical with physical design. The Figure was prepared by using Power Soft's Warehouse Architect 6.0. The key of a dimension table is marked with an underline. Sales-Fact table's key is a concatenated key composed of keys in each dimension table. This star schema consists of five tables: one Sales-Fact table, and four dimension tables (Time, Promotion, Product, and Store).

After the development of DW, data is extracted from operational database, is transformed and loaded to

DW. Based on the Transformation & Mapping information, data is actually transformed and loaded by using Info Pump 2.1, an extracting component.



<Figure 3> Star Schema

DW application is developed by Star Tracker Solo access tool, one of the application components. Another access tool is developed by using MS Visual Basic. Through this application system, business end-users can search and analyze DW under canned or ad hoc queries. Canned queries are illustrated in Table 6. The application information table also shows present queries or reports, their explanation, their estimated time, and user number. For instance, Product Sales Report/Query compares the product sales amount of last and this year in detail.

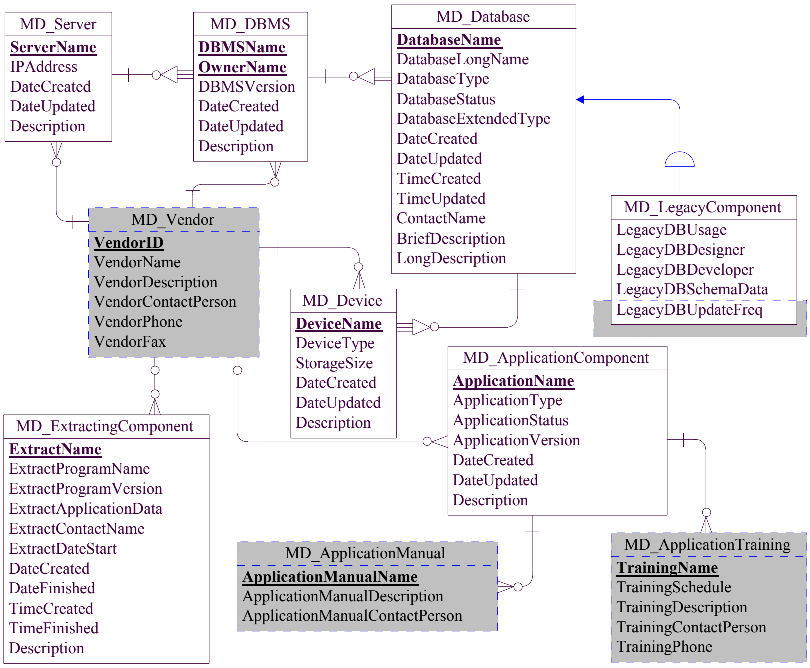
〈Table 6〉 Application Information

Details / Report/Query Name	Explanation	Estimated Time	User Number (Primary Department)
Product Sales -- This Year vs. Last	- Compare this year's product sales to last year's sales - Further, analysis by store	0:01:20	5 (Product Department)
Midweek vs. Weekend Purchase Behavior	- Analyze customer behavior for midweek vs. weekend - Further, usage pattern by promotion	0:00:45	4 (Sales Department)
Weekend Sales Behavior By Promotion	- Find the most effective promotion for weekends - Further, promotion development	0:01:00	6 (Two Department)
Midweek Sales Volume By Store	- Select stores with highest and lowest midweek sales volume - Further, investment in region and advertisement	0:02:05	4 (Two Department)
Gross Profit By Sales Region By Display Type	- Find the most effective display type by region - Further, display type analysis	0:00:35	6 (Two Department)

4. Metastore Development Phase

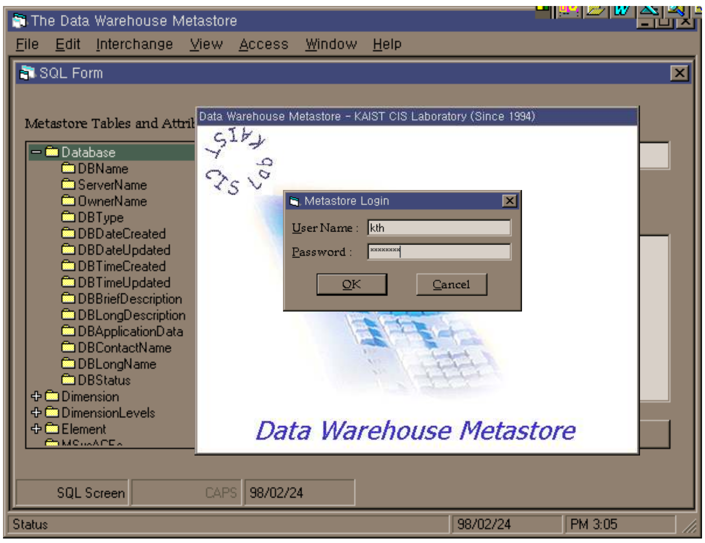
Domain-specific metadata are added to core metadata schema. Overlapped information such as Application Description attribute (information overlapped with Application Description) is deleted from the final schema.

Figure 4 shows a part of final metadata schema. The figure is a conceptual model. The dotted-line gray rectangle expresses the added entities and attributes. For instance, in MD_Legcy Component entity, Legacy DBUpdateFreq attribute is also an additional attribute. After such tables, attributes, and relationships are added, the final metadata schema is produced. After this process, metastore is implemented by using Microsoft SQL Server 6.5 on Windows NT 5.0 operating system.



〈Figure 4〉 Partial Schema of Final Metadata Schema

Figure 5 shows a metastore prototype system. The system is used in exploring and managing metadata in a grocery store case. The system was developed by using Microsoft Visual Basic 5.0.



〈Figure 5〉 Metastore Prototype System

5. Conclusions

In today's information processing, a data warehousing system plays an important role in integrating enterprise information. Especially, the metadata component is a key to successfully implementing, managing, and using data warehousing system. Most DW development methodologies have not considered metadata development simultaneously; there is an increasing necessity to adopt an DW development methodology which develops a DW and its metadata simultaneously. The metadata must be integrated with data warehousing system. Without the metadata, end-users may depend on their own knowledge or ask technical users for information about data at any time. That is, the decision support of DW is under the control of technical users.

The proposed methodology also points to a stage where metastore system can be effectively developed. In particular, by developing other components on the basis of metadata, rather than purchasing metadata tool and managing only the metadata, a consistent development environment is created. Non-metadata component development on the basis of metadata is far more useful when developing a data warehousing system with a low budget.

The major contributions of this paper can be summarized as follows: to propose a methodology called a DW development methodology for developing a data warehousing system on the basis of metadata component development. The methodology can be effective for developing a data warehousing system with metadata, not for adopting commercial products.

References

Anahory S. and Murray, D., *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, Addison Wesley, 1997.

Atre, S. and Storer, P., "Data Distribution and Warehousing," *DBMS*, Vol. 10, 1995, pp. 54-62.

Bischoff, J. and Alexander, T., *Data Warehouse: Practical Advice from the Experts*, Prentice Hall, 1997.

Bouziane, M., "Metadata Modeling and Management," *Ph.D Dissertation, Rensselaer Polytechnic Institute*, 1991.

Chaudhuri, S. and Dayal, D., "An Overview of Data Warehousing and OLAP Technology," *SIGMOD Record*, Vol. 26, No. 1, 1997, pp. 65-69.

Denzer, R. and Guttler, R., "Requirements of the Meta Information Models for the Environmental Domain," *Journal of Computing and Information*, Vol. 2, No. 1, 1996, pp. 1328-1335.

Devlin, B., *Data Warehouse from Architecture to Implementation*. Addison Wesley, 1997.

Froeschl, K.A., *Metadata Management in Statistical Information Processing: a Unified Framework for Metadata-based Processing of Statistical Data Aggregates*, Springer-Verlag, 1997.

Gill, H.S. and Rao, P.C., *The Official Guide to Data Warehousing*, Que, 1996.

Golfarelli, M., Mao, D., and Rizzim, S., "Conceptual Design of Data Warehouses from E/R Schemes," *Proc. of the 31st IEEE Annual HICSS*, 1998.

Gray, P. and Watson, H.J., *Decision Support in the Data Warehouse*, Prentice Hall, 1998.

Griffin, J., "Metadata: Capturing the Heart of the Data Warehouse," *Application Development Trends*, 1997.

Hackney, D., *Understanding and Implementing successful Data Marts*, Addison Wesley, 1997.

Hoven, J.V.D., "Data Warehousing: New Name for the Accessibility Challenge," *Information Systems Management*, 1997, pp. 70-72.

Inmon, W.H. and Hackathorn, R.D., *Using the Data Warehouse*, John Wiley, 1994.

Inmon, W.H., *Building the Data Warehouse*, John Wiley, 1993.

Inmon, W.H., Imhoff, C., and Battas, G., *Building the Operational Data Store*, John Wiley, 1996.

Jarke, M., Jeusfeld, M.A., Quix, C., and Vassiliou, P., "Architecture and Quality in Data Warehouses: an Extended Repository Approach," *Information Systems*, Vol. 24, No. 3, 1999, pp. 229-253.

Kimball, R., *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*, John Wiley, 1996.

Kutsche, R. and Sunbul, A., "Meta Data Support for Evolutionary Software Systems," *the IEEE 18th SCCC '98*, 1998, pp. 84-90.

Laudon, K.C. and Laudon, J.P., *Management Information System: Organization and Technology*, Third Edition, Macmillan College Publishing Company, 1994.

McClure, C., *The Three Rs of Software automation: Re-engineering, Repository, Reusability*, Prentice Hall, 1992.

McFadden, F.R. "Data Warehouse for EIS: some Issues and Impacts," *Proc. of the 29th IEEE Annual HICSS*, 1996, pp. 120-129.

Moriarty, T. and Mandracchia, C. "Heart of the Warehouse," *Database Programming and Design*, 1996, pp. 70-74.

Murtaza, A.H., "A Framework for Developing Enterprise Data Warehouses," *Information Systems Management*, 1998, pp. 21-26.

Noaman, A.Y. and Barker, K., "Distributed Data Warehouse Architectures," *Journal of Data Warehousing*, Vol. 2, No. 2, 1997, pp. 37-50.

Poe, V., *Building a Data Warehouse for Decision Support*, Second Edition, Prentice Hall, 1997.

Silverston, L., Inmon, W.H., and Graziano, K., *The Data Model Resource Book: a Library of Logical Data Models and Data Warehouse Designs*, John Wiley, 1997.

Subramanian, A., Smith, L.D., Nelson, A.C., Campbell, J.F., and Bird, D.A., "Strategic Planning for Data Warehousing," *Information and Management*, Vol. 33, 1997, pp. 99-113.

Tannenbaum, A., *Implementing a Corporate Repository*, John Wiley, 1994.

Thomsen, E., *OLAP Solutions: Building Multidimensional Information Systems*, John Wiley, 1997.

Timpf, S., Raubal, M., and Kuhn, W., "Experiences with Metadata," *the 7th Int'l Symposium on Spatial Data Handling*, 1996, pp. 12B.31-43.

Tozer, G., *Metadata Management for Information Control and Business Success*, Artech House, 1999.