'98 International Conference Digital Library & Knowledge
September 22, 1998, Seoul Korea

A Metastore-based Data Warehouse Development Methodology

Heeseok Lee and Taehun Kim

Graduate School of Management, Korea Advanced Institute of Science and technology, 207-43 Cheonyang, Dongdaemoon, Seoul 130-012, South Korea

Abstract

Data warehouse (DW) is important for analytical processing. Metadata is a key to its architecture. This paper proposes an architecture that consists of seven components. To illustrate data warehouse environment (DWE), this paper proposes taxonomies having four flows. On the basis of the taxonomies and metadata, this paper proposes a methodology for building the data warehouse and metadata simultaneously. This integrated development methodology (IDM) consists of seven phases: (i) preparatory phase, (ii) requirement analysis phase, (iii) data warehouse development phase, (iv) operational data store development phase, (v) data mart development phase, (vi) metastore development phase, and (vii) maintenance phase. A metastore system is proposed to help develop metadata interactively. An illustrative example is investigated to demonstrate the usefulness of IDM.

1. Introduction

In the domain of databases, a new buzzword, the "data warehouse (DW)" has lately appeared. DW technology has progressed quickly, and such new terms are constantly being introduced. However, little academic research has been conducted in this area [38]. Information technology strategists have been primarily grappling with the "what" issues of data warehousing and are only now considering the "how" issues [1]. We now need to take an academic look, rather than a vendor-oriented glance, at DW.

Inmon, the "father" of DW, defined it as "a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision making process" [25]. But data warehousing is merely a new name for an old idea-that of making an the enterprise's data more accessible [23]. Thus DW is supplied as a fundamental architecture for enterprise information processing.

The technologies of DW are inherited from those of general databases-i.e., distributed databases such as data mart (DM). Though this heritage, DW technology is perceived as a newcomer by most information technology professionals, because of the environmental differences between DW and operational databases. The latter are based on on-line transaction processing (OLTP) applications supported by the operational database, which continuously produces data; whereas the data warehouse environment (DWE) is based on on-line analytical processing (OLAP) applications, in that DW continuously supplies integrated and analytical information to business users. The term OLAP has been invented in recently to represent the alternative to OLTP [12].

Up to the present, there is a primitive stage in the DW technologies. In investigating DW there are two important factors to consider: one is to develop DWE based on the decision making of business end-users, and the other is to develop metadata within it.

Methodologies to develop DW, DM, and operational data store (ODS) separately have had inconsistent results in integrating of scattered and independent databases. Most existing methodologies can be categorized as either simple, perfect, or point solution [31]; however, such approaches have many shortcomings. The best fit to eliminate these shortcomings is an evolution solution; the DWE must evolve as corporate and business evolves. In considering aspects of expansion in the DWE such as adding one

ODS and/or two DMs to one DW, simply connecting them may result in complexities and inconsistency of function. Besides, from the viewpoint of solution, there is no clear idea which elements such as software, hardware, and data fields should be combined; a new environment, which is inefficient in respect of time and cost, might be developed. As the evolution solution covers all elements to be added systematically and iteratively, so the DWE continues to evolve. Composing flexible and expandable architecture is the most expedient step to DWE; besides, separately developed methodologies have the problem of inconsistent metadata and have not been the part of its development. Metadata is generally considered as software in legacy methodologies, so there is an increasing necessity to develop an integrated development methodology (IDM) for DW.

Metadata is a key to success of DWE, but metadata management has failed precisely because metadata management was segregated from the development process [26]. Metadata must be directly included in the design of any information system being built, and the system must offer an interface to access information from outside [11]. The metadata must be developed along with the DW and/or the DM and/or the ODS. Metadata is useful in several aspects of a data system, including data access, data management, and data analysis [13]. This paper concentrates on the first two of these. The metastore system supplies business end users and technical users with access to metadata and serves as a repository to integrate and synchronize the various metadata sources.

The objectives of this study are as follows: (i) to define and enumerate DWE; (ii) to define metadata in DWE, establish standard for metadata, and develop metadata schema; and (iii) to present the IDM phases and develop metastore system to interactively support metadata development while simultaneously supplying metadata applications such as browsing and searching, with its methodology.

2. Data Warehouse Environment

2.1 Taxonomy of Data Warehouse Architecture

The data warehouse architecture (DWA) can be graphically represented by two taxonomies which present five flows. Hackathorn [19] presented five such primary information flows in DWA: Inflow, Upflow, Outflow, Downflow, and Metaflow. Inflow, Outflow, Upflow and Downflow are called Dataflow; Upflow and Downflow are also termed DWflow as they occur in DW itself. Table 1 enumerates and explains the five flows.

Table 1 The five flows and explanation

Flows				Explanation
Five Flows	Dataflow	Inflow		Legacy data, as well as data from diverse sources, and external Databases, is cleaned up and fed into the DW.
		DWflow	Upflow	Highly detailed data is aggregated, summarized, and otherwise processed in the warehouse to make it easier for users to get quick responses to their queries.
			Downflow	From becoming turgid with marginally useful old data, it has to send older data off to an archive, where it can be accessed if needed.
		Outflow		Users run both packaged and ad hoc queries to get the business information they need.
	Metaflow			Both users and IS need to know about the data in the warehouse. That information comes from several flows of metadata.

In this Table, Outflow is defines as data flows from DWE to business end-users. These five flows can be used to express all the processes in DWE in taxonomies.

Dataflow is represented by Fig. 1. In Fig. 1, the simplest format is the combination of a lazy pattern in Inflow, a single pattern in DWflow, and an access pattern in Outflow. The ultimate purpose of DWE is to extend to evolve to combinations of hybrid, triple, and data mining patterns. Selecting a point along

each arrow results in affecting the data warehouse architecture and infrastructure (DWAI).

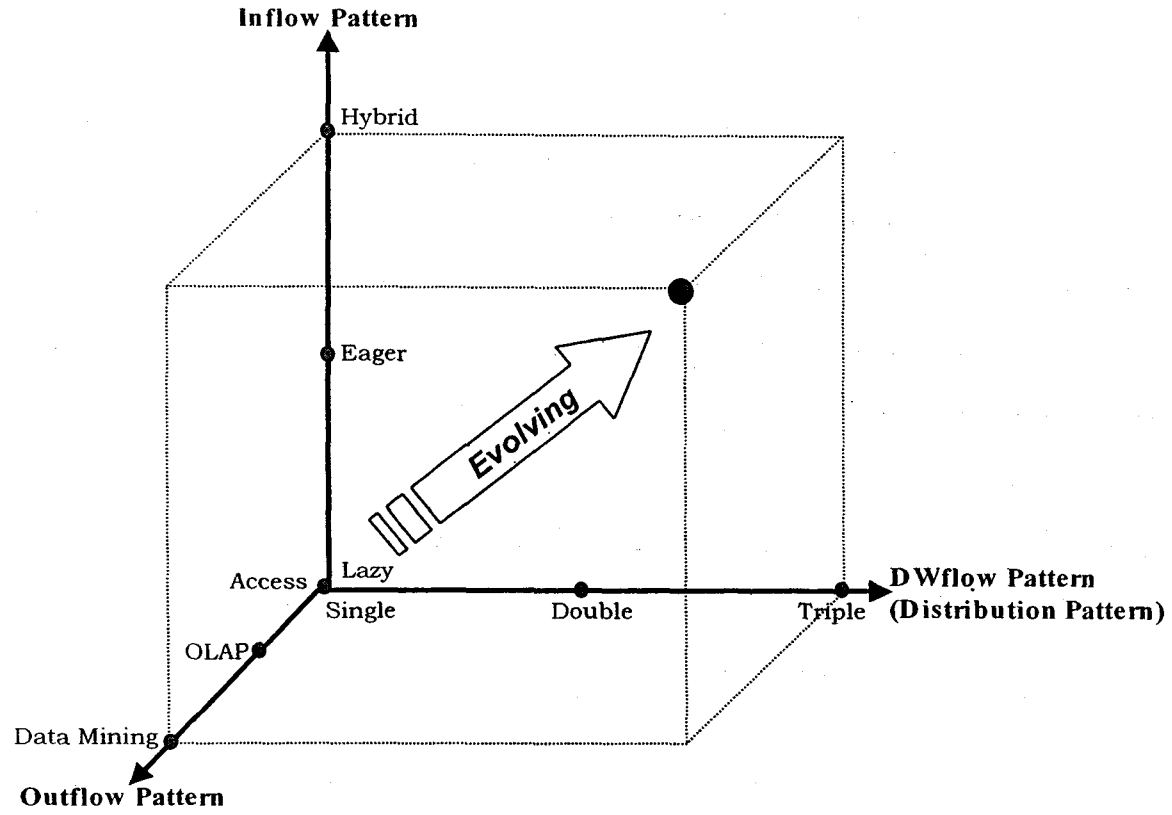

Fig. 1. Taxonomy of Dataflow

The Inflow pattern expresses how to access legacy systems; the lazy or demand approach allows end-users to directly access legacy databases [58]. For this reason, it is called virtual DW [44]. The eager or in-advance approach is commonly referred to as the data warehousing approach [58], or the informational DW approach [44]. In hybrid approach, some information is stored in a DW while other information is retrieved on a demand approach [58]. That is, both the lazy approach and the eager approach are incorporated. This approach is composed of ad-hoc queries and reports as a part of lazy and canned-queries and reports as a part of eager.

The DWflow pattern that expresses how to distribute processing database as a degree of data distribution is explained and summarized in Table 2.

Table 2. The comparison of DWflow pattern

Type	ODS	DW	DM	Suitable Environment
None	—	—	—	Not suitable for DW environments. Burdensome to legacy system.
Single	✓	—	—	Suitable to short-term decision support. Mainly for operational data.
	—	✓	—	Standard form. Includes parts of business enterprise. This form is called as the "prime" DWA.
	—	—	✓	Suitable for departmental business units or business functions. Easier management. In this case, it is called Independent DM.
Double	✓	✓	—	Suitable for supplying both operational data and decision support data enterprise-wide.
	✓	—	✓	Suitable for operational data and functional or departmental unit data but it is *impossible*.
	—	✓	✓	Suitable for distributed database environment. Enterprise-wide views and departmental view.
Triple	✓	✓	✓	The most complex and ideal form. Operational and distributed information for decision-making. This form is known as the "nirvana" DWA.

Complexity ▼

Because the "None " type is not truly DWE, it is not suitable. The foundation of the environment is the "Single" type, which adopts one database among ODS, DW, and DM. On the basis of the single type, DWflow pattern could be developed into a more complex type: the "Double" type and the "Triple" type. In the double type, DMs are discussed contain no ODS information, unlike DWs [5]. The double and triple types in the database sequence are discussed in the following section, "Data Warehouse Architecture".

Outflow is explained and summarized in Table 3. The simplest approach is by means of Access tool, and the most complex and expensive, but efficiently functional approach is by means of a data mining tool. In general, access tools are similar to query tools and report writers [47]. Categories of the OLAP tools include DOLAP (Desktop OLAP), ROLAP (Relational OLAP), MOLAP (Multidimensional OLAP), and HOLAP (Hybrid OLAP) [15, 21]. The data mining tool does the discovery and gives information instead of gives information a question, whereas access tools and OLAP tools return records that satisfy a previously formulated query [43].

Table 3 The comparison of the Outflow pattern

Approach	Access	OLAP [56]	Data Mining [56]
Primary Objective	Simple Query & Reporting	Hypothesis testing Data exploration Multidimensional view	Discovery
Driven	Schema-driven	User view driven	Data-driven
Interactive	Interactive	Interactive (User-directed)	Batch
Views	Top-down	Top-down	Bottom-up
Static/Dynamic	Static (Ex. Provide result set in tabular format)	Static (Ex. provide result set)	Dynamic (Ex. provide predictive model)

Complexity

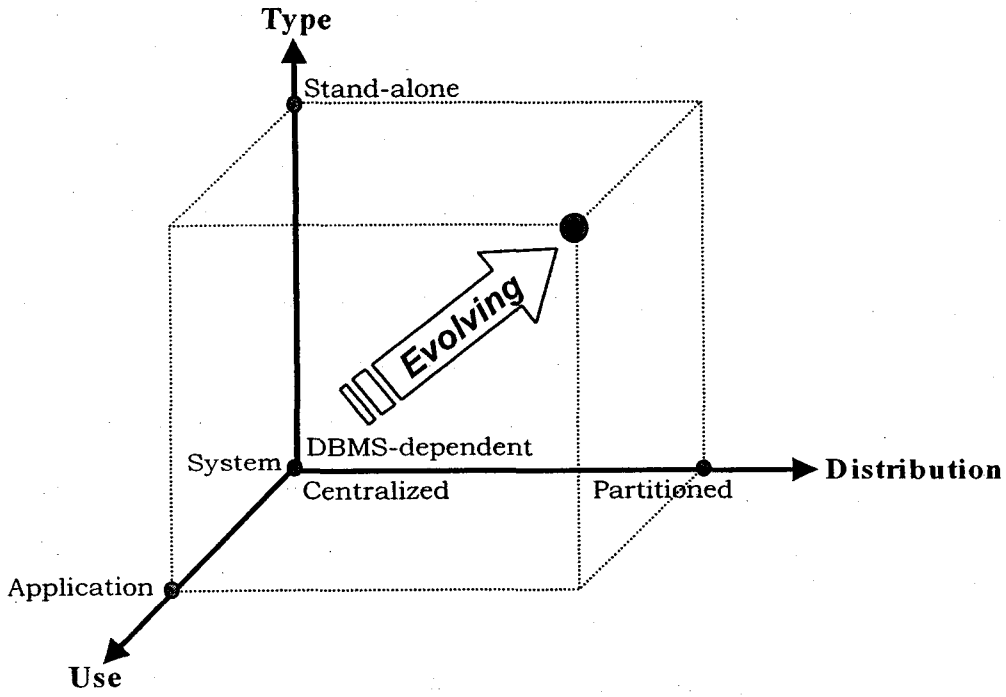

Fig. 2. Taxonomy of Metaflow

Metaflow is diagrammed in the shown in Fig. 2; this taxonomy is dependent on and varies with the DWA. As in the case of Dataflow, selecting a point of each arrow affects the DWAI. The Metaflow taxonomy contains three arrows: distribution, type, and use.

In the "distribution" pattern, many of the tools and problems generally associated with data distribution are also relevant to the DWs [1]. The metadata may be maintained centrally at one site, or in

a distributed fashion [45]. Therefore a centralized metadata database may be containing the all information about the all databases related to the DWA or many partitioned metadata databases fed a centralized database may be containing all the information and each information about each database. In the latter case, the ODS has a metadata database having the information about it, so do the data warehouse and the DM. The centralized metadata database is all its information about the all the databases and metadata distribution. But the right of arrow, the **partitioned distribution** is the economic and ideal than the centralized and partitioned distribution that is omitted in Fig. 2.

The second pattern is as to "type". The stand-alone system can support one or more database management system (DBMS) through interfaces. Such a system enables the data administrator to exercise tight over the DBMS. A dependent system is more often than not a commercially available package, designed for general-purpose. Duyn [14] explained that whereas the stand-alone system has great flexibility and portability, the dependent system is restricted to a particular DBMS. And whereas the dependent system can be useful in a situation where all operational data is stored under a single DBMS, in multiple DBMSs or other situations, the stand-alone system may be the only that can provide centralized control of all data. Thus if there are many databases in DWA, the stand-alone system is the best fit to it.

The final pattern is as to "use". The metadata can be classified into business metadata (or application metadata) and technical metadata (or system metadata). The technical metadata contains technical information about warehouse data and is intended primarily for use by warehouse administrator and the business metadata contains information that provides end-users with an easy-to-understand business perspective of the warehouse data [57]. The DWE itself is mainly devised and used for business end-user. Thus the metadata database contents must contain the more business metadata than the system metadata, though the distinction between the business and the technical metadata is not clear and concrete.

2.2 Data Warehouse Architecture

Under the legacy OLTP environment, decision support has been carried out by lazy approach in

Dataflow taxonomy without a DW. Also most of the legacy architectures from the literatures are used as the prime DWA. According to the taxonomy, the eager approach having single DW and simple access tool are used in most architecture without considering evolving factors.

For example, the architecture has not considered the ODS and bottom-up approach of DMs in Kosar's architectural framework [35], and it is expressed as double approach with a distribution by the Dataflow taxonomy without considering the Metaflow taxonomy. Also, Chaudhuri and Dayal's architecture [9] is same with the above except for considering the metadata repository storing the metadata. Against the two cases, the DWE in this paper covers and pays regard to the evolving architecture from the prime DW to the nirvana DW according to the taxonomies representing Dataflow and Metaflow.

The DWE includes data warehousing system, business end-user, customer, and developer. Fig. 3 shows the DWE and seven components. The DWAI correspond to elements of data warehousing system. The warehousing system consists of input-process-output components. The input part is operational database and its software, the extracting software. The process component is the database region, informational database including the DW, ODS, and DM components. The output is the area of utilizing process, that is application software. In control of the other parts is the metastore system, including the metadata component.

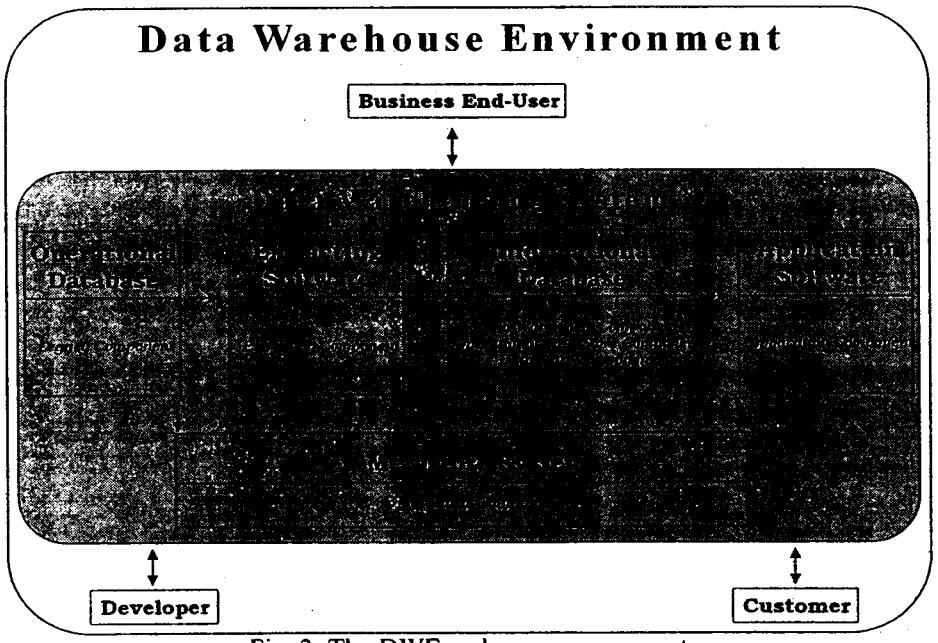

Fig. 3. The DWE and seven components

The DW's focus is neither application, nor infrastructure, but architecture which may be transformed and expanded. The DWA is a special kind of client-server architecture. Poe [47] presented DWA has as a primary component, a read-only database used for decision support. The DWA's fundamental characteristics are determined by the DWE. They are relations that the legacy component feeds on informational databases, that between legacy component and informational databases there must be tight- or loose-coupled, that between informational databases there must be tight- or loose-coupled, and that application component works to informational database. The data warehouse infrastructure (DWI) refers to the platforms, database management systems, gateways, networks, front-end tools, as well as training in these technologies and other components necessary to make the architecture function; they are closely related to architecture [46]. Thousands of the different infrastructures can be derived from the same architecture.

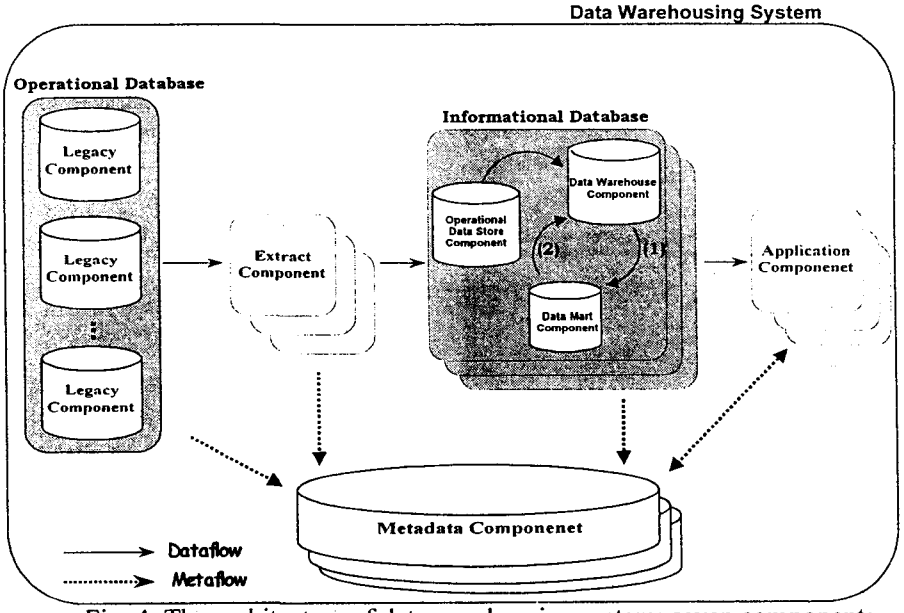

Fig. 4. The architecture of data warehousing system: seven components

Fig. 4 illustrates the architecture of the data warehousing system. In Fig. 4, cases where all components are present that is, the most complex architecture, are called the nirvana DWA. Nirvana architecture is composed of seven components: legacy, extract, ODS, DW, DM, metadata, and application. The legacy, extract, metadata, and application components are indispensable. The database component is also indispensable, and is selected by combination of each database component. The strategy to use informational database must be referred to Table 2 used to indicate combination of

database component arrays. The arrow expresses the orders of database arrays. In the case of one ODS and one DW of the double type, the ODS information is fed into DW information.

Also, number (1) indicates top-down approach and number (2) indicates bottom-up approach in relation between DW(s) and DM(s) [36]. The top-down has advantages that supply planning blueprint and consistent method. It has disadvantages of many time and costs. The bottom-up approach has advantages and disadvantages vice versa of top-down approach.

In Fig. 3 and Fig. 4, whereas the extract and the application component are a part of software program, legacy component, informational database, and metadata component are a part of database. Table 4 shows the relative comparison and interrelationships of databases that are the legacy component, the informational database, and the metadata component. Each database is used in the different environment where operational database mainly acts on OLTP environment, whereas DW and DM mainly on OLAP. In the case of summary information, what calculate high currency data came from operational database and present dynamic summary is used in ODS, while in DW and DM it calculate lower currency data and store summary information. So in these two databases there is low currency, integrated and historical data.

Table 4 Comparison and interrelationship of databases

Criteria	Operational Database	Informational Database			Metastore
	Legacy Component	ODS Component	DW Component	DM Component	Metadata Component
User Number	Most	Many	Few	Fewer	More
Environment	OLTP	OLTP/OLAP	OLAP	OLAP	Information Navigation
User Community	Non-Management Level	Non-Management/ Management Level	Management Level	Department Level	Non-Management /Management/ Department Level
Summary	Few	Dynamic Summary	Stored Summary	Stored Summary	Few
Dynamics of Updates	Higher Currency	High Currency	Low Currency	Low Currency	Higher Currency
Data	Unintegrated Perspective	Integrated, Corporate Perspective	Integrated, Historical Perspective	Integrated, Historical Perspective	Integrated, Historical Perspective
Decision-Making	Very Short-term	Short-term, Corporate	Long-term	Long-term, Departmental	Helping Users in Department/ Corporate
Interrelation	Feeding ODS/DW/DM	Feeding DW	Feeding DM	Feeding DW	Receiving all the database information

3. Integrated Development Methodology

3.1 Motivation

Four major strategies used for building the corporate DW are simple solution, point solution, perfect solution, and evolution solution. The simple, point, and perfect solution have shortcomings [31]. The evolution solution corrects shortcomings of above three solutions and has a flexible structure. Inmon and Hackathorn [25] explains that DW is an evolutionary process, not a revolutionary process. The comparison for these solutions is shown in Table 5. From the viewpoints of evolutionary approach, the point solution is equal to bottom up approach and the evolution solution is equal to top down approach.

The evolution solution in evolutionary approach has the more robust, flexible, and omnipresent environment in which DWE is used, than the other solutions. Besides, this solution views and approximates enterprise-wise information management at the corporate-level. Thus this solution having iterative process can insert additional elements into the environment, while this method is not ending.

Table 5 Comparison of solutions related to build the DWE

Solution		Environment	Approach	Issue	Shortcomings
One-Shot Approach	Simple Solution	Naive	Data Transfer	Replication	Not Successful
	Perfect Solution	Brittle	Corporate	Over-engineered	Not Possible
Evolutionary Approach	Point Solution	Reactive	Subject Area	Isolation	Not Durable
	Evolution Solution	Reasonable	Corporate	Additional Elements	Not Ending

Poe [47] proposed decision support life cycle (DSLC). Though the goals and the data structures of on-line transaction based systems and decision support systems are different, DSLC does not deviate from the classical the system development life cycle (SDLC) that begins with the gathering of end-user requirements and ends with implementations. But the SDLC method is ill-suited to decision-oriented applications [37]. The systems life cycle is still used for building large OLTP systems. Decision-making

can be rather unstructured and flexible. Requirements constantly change or decisions may have no well-defined models or procedures. Decisions makers often cannot specify their information needs in advance. They may need to experiment with concrete systems to clarify the kinds of decisions they wish to make. This high level of uncertainty cannot be easily accommodated by the SDLC. Thus, prototyping is particularly true of decision-oriented, where requirements tend to be very vague. But prototyping is inappropriate for large complex systems [37]. Thus, in such as the DW, mixtures are necessary for iteration of user evaluation and large complex systems and projects. The sequential waterfall has met with mixed success [54]. It has both the iterative and spiral approaches with SDLC.

Besides, metadata management that is a key to success in the DW is seldom developed with the DW development in the past research. Simply relying on the metadata tool and absence from methodologies causes it to look down on the importance of metadata, and results in failing metadata management.

In this paper, by using the preparatory phase to find DW candidates, it covers weakness of applying SDLC to DWE. Also this methodology adds iteration approach and evolution solution in the IDM with developing metadata together.

3.2 Integrated Development Methodology

Fig. 5 shows the IDM of DWE. As the Fig., IDM is composed of seven phases: preparatory phase, requirement analysis phase, DW development phase, ODS development phase, DM development phase, maintenance phase, and metastore development phase.

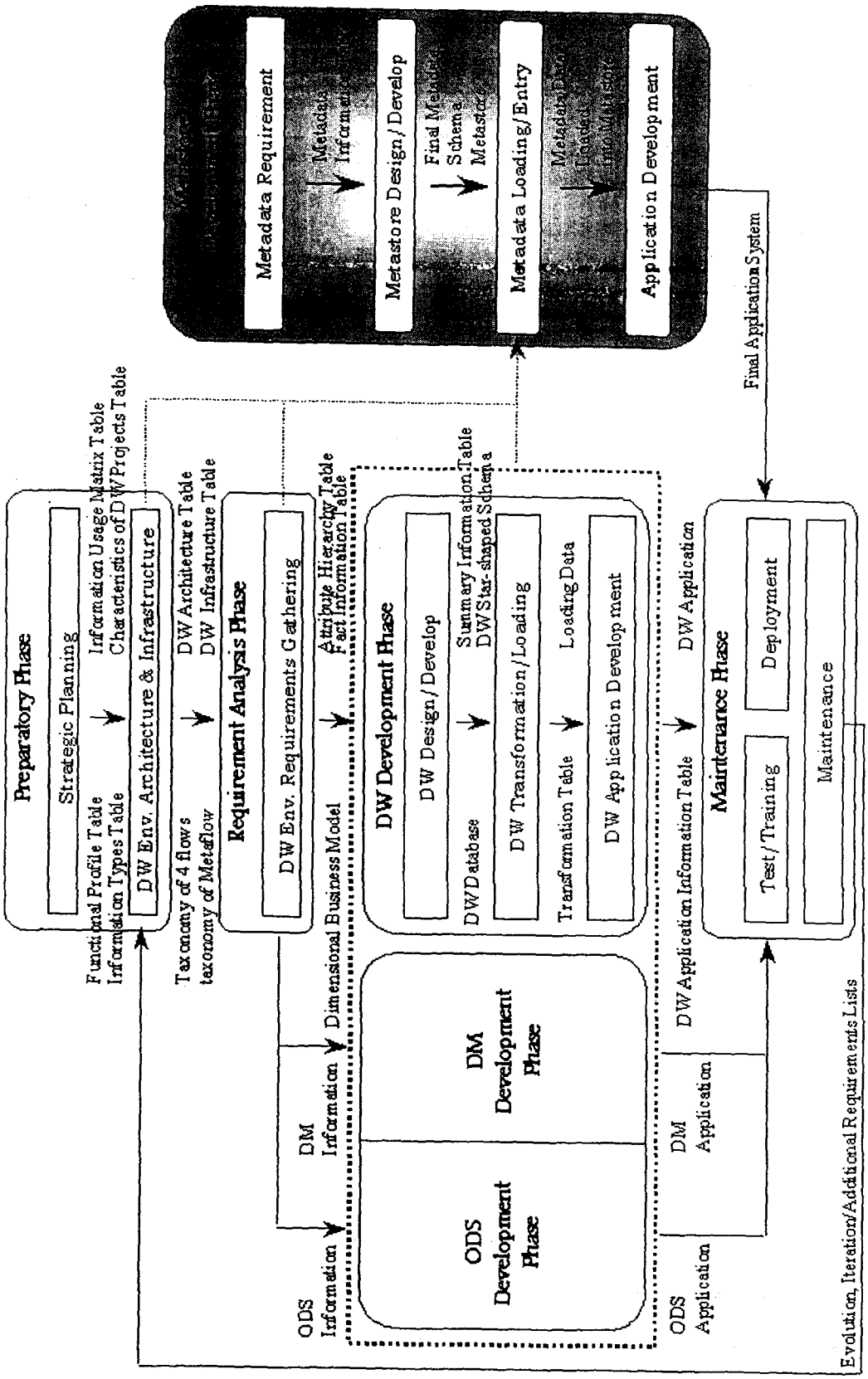
Preparatory Phase
Strategic Planning
Functional Profile Table
Information Types Table
Information Usage Matrix Table
Characteristics of DW Projects Table
DW Env. Architecture & Infrastructure
Requirement Analysis Phase
DW Env. Requirements Gathering
Taxonomy of 4 flows
taxonomy of Metaflow
DW Architecture Table
DW Infrastructure Table
Dimensional Business Model
Attribute Hierarchy Table
Fact information Table
DW Development Phase
DW Design / Develop
DW Database
Summary Information Table
DW Star-shaped Schema
DW Transformation / Loading
Transformation Table
Loading Data
DW Application Development
DW Application
Maintenance Phase
Test / Training
Deployment
Maintenance
DW Application Information Table
Metadata Requirement
Metastore Design / Develop
Final Metadata Schema
Metastore
Metadata Loading / Entry
Application Development
Final Application System
ODS Information
DM Information
ODS Development Phase
DM Development Phase
ODS Application
DM Application
Evolution, Iteration/Additional Requirements Lists

Fig. 5. The IDM

The preparatory phase is composed of two stages: strategic planning stage and DWE architecture and infrastructure stage. The strategic planning stage is a process of finding the data warehousing candidates and determining the contents. In this stage [52], there are outputs of four tables: (i) functional profile table which describes the mission and functions of the department and its subdivisions, (ii) information types table which describes the types of information used or generated by the department, (iii) information usage matrix table which describes the associated computing platform or storage medium and the functions supported by the information, the transmission of the information to or from other departments or external organizations, and the currency of the various types of information, and (iv) characteristics of DW projects table which describes the list of recommended the projects or initiatives. The purpose of this stage is ultimately to find candidates moderate to DWE subjects for business end-users have no ideas about what they need, and further analysis for the world of tomorrow. These output tables of this stage might come to next stage, DWE architecture and infrastructure stage that defines architecture and E followed by Dataflow taxonomy and Metaflow taxonomy in DWE.

The DWE architecture and infrastructure stage has four parts: (i) taxonomy of four flows in Fig. 1, (ii) taxonomy of Metaflow in Fig. 2, (iii) DWA table which describes and summarizes components selected among Fig. 4, and (iv) DWI table which describes and summarizes technical information corresponding to its architecture information. This stage suggests frameworks, outlines, and scopes of the project. The outputs came from the preparatory phase as the starting point of evolutionary and iterative process, go to the requirement analysis phase with the entire view of the DWE

Requirement analysis phase that discerns users' need and business modeling has three parts: (i) dimensional business model that shows metrics, dimensions, and relationships which can easily be presented back to business users for their verification [47], (ii) attribute hierarchy table, and (iii) fact information table.

The dimensional business model could be a graphical description of the simple star schema, and in that case the logical model and physical models will be identical with the exception of physical table partitioning [47]. Instead of dimensional business model the traditional logical model such as the development of entity relationship diagrams is able to applied to DW development [47]. But if possible,

using the dimensional business model is the best way in DW development, though there is a semi-automated methodology from entity-relationship (ER) schemes to fact schemes [17], because there is no simple way to change it, even when modeling the same data [33]. The attribute hierarchy is a parent-child relation between attributes within a dimension. The Hierarchies are attributes of dimensions [55]. The attribute hierarchy table describes the attribute hierarchy and summarizes backbone stem and relative which expresses the same class within one stem stage. This relative does not exist legacy structures such as multidimensional structuring [56] as new views to the hierarchy. The fact information table describes the information of additive variables and semi-additive variables.

There are two kinds of informational processing that may be referred out from the legacy systems environment- ODS informational processing and DW informational processing [25]. Also DW informational processing can be divided into DW part and DM part. Therefore, the requirement phase is branched into three categories: ODS requirement, data store requirement, and DM requirement. These requirements may flow into ODS or DM or DW development phases.

The DW development phase, in which dimensional business model is converted into the physical DW database models including aggregations and partitions, physically implementing the designs and developing the applications, is composed of three stages: (i) DW design/develop stage, (ii) DW transformation/loading stage, and (iii) DW application development stage. The DW design/ develop stage has three outputs: (i) summary information table including derived variables and aggregations, (ii) DW star-shaped schema such as star or snowflake or mixed schema, and (iii) DW database which is physically implemented. DW transformation/loading stage, which is the process of transforming legacy data and loading data into DW database has two outputs: (i) transformation table which represents relationship with source information to DW information, and (ii) Loading data into DW. The transformation table includes criteria, cardinality of legacy attribute to DW attribute, variable, and source information. The table has five criteria: data conditioning, integration, stasis, simplification, and amplification [41].

DW application development stage under the developed DW database has two outputs: (i) DW application information table including information of canned queries and reports, and (ii) DW

application developed by development softwares or DW-specific application softwares.

The process of ODS and DM development phase are similar to that of the DW development except for different domain and purpose [10, 27]. Also the similar applications are derived through its phases.

The metastore development phase into which information came from the prior five phase flows and that is developed with their phases is composed of four stages: (i) metadata requirement, (ii) metastore design/develop, (iii) metadata loading/entry, and (iv) application development. Metadata requirement stage which takes needs through business users' interviews has output of metadata information table that represents domain-specific and additive metadata in comparison to core metadata schema. Metastore design/develop stage modeling physically metadata and implementing its database metadata data has two outputs: (i) final metadata schema which core metadata schema pluses domain-specific and additive information, and (ii) metastore which is database storing metadata. Metadata loading/entry has metadata data loaded into metastore. The arrows entered into metadata loading/entry stage means that the information outputs of each stage go to the its stage as input data. The application development has a final application system that searches and browses metadata in metastore.

After all the development phase, maintenance phase follows the next step. The maintenance phase is composed of three stages: (i) test/training including of testing access tools and application systems, validating data, and proactively supporting users (ii) deployment [47], and (iii) maintenance [22]. Through this phase, it is considered whether to evolve DWE. The change of the DWE such as a new DM, or users' needs of the new-incoming requirements such as new sources, and queries causes it to iterate the evolutionary development of the DWE.

3.3 Metastore Metadata

3.3.1 Warehouse Metadata

As the DWE evolves, also the metadata must be evolved and extended. On the contrary, the metadata of the operational database environment in comparison with the DWE is in sluggish state for OLTP

characteristic. But as respect of integration, the distributed database must be managed by the integrated metadata. Metadata has been around since the beginnings of the computer age [18]. Metadata has been in all the database fields. The domain to where metadata is applicable and interrelated to its components are workflow design and management, document management, computer-aided design, application development, and so on [3]. The traditional metadata used by database administrators to manage and maintain internal tables and other structures in the database [24]. The metadata in the operational environment are mainly devised for on-line transaction and for technicians such as database administrator and developer, not business users [6].

But in the DWE, not only technicians but also business end-users can touch metadata. Rather than technicians, the end-users surf the metadata for decision support. The metadata is surely a core of the DWE because its environment focused the end-users decision-making. Because the operational metadata under the OLTP environment mainly is used by technical users other than warehouse metadata, system-oriented metadata is stored. Historical metadata information is piled up the warehouse metadata under user centric OLAP environment. Table 6 shows the comparisons with the general operational metadata.

Table 6 Comparisons with the general operational metadata

Criteria	Operational Database Metadata	Informational Database Metadata
Time	Few/A Few Versioning	Versioning-Historical Information (Granularity)
Users	Small Number of Users (Developer>End-Users)	Large Number of Users (Developer<End-Users)
Scope	Narrow Functional Scope	Enterprise-wide Scope
Environments	OLTP, Tool-centric	OLAP, User-centric
Using Aspect	Efficiency use of Legacy (Managing) /Optional	Effectiveness use of DW (Searching and Managing) /Mandatory
Data Structure	Unstructured/Structured Format	Structured Format
Data Type	Text/Video/Graphic/Sound/Unstructured	Text/Video/Graphic/Sound/Unstructured
Technology	Relational/Object Technology	Relational/Object Technology
Generation	Automation :System-level	Automation/User-defined(Manual): System-level/application- level
Integration	Unintegrated	Integrated
Understanding	Legibility-difficulty	Legibility-easy
Structure Change	Non-Changing	Changing
Distribution	Centralized/Distributed	Centralized/Distributed

3.3.2 Metadata Schema

The data warehousing and knowledge management fields rediscover what the early decision support system (DSS) field knew: It's information use, not information supply, that we need to address and encourage and data supply doesn't lead create information [30]. The metadata helps data to become information and business end-users to smoothly use a DWE. The objective of metadata is to give an object (data) a *name*, that is "meaning". As result, it is an information. The information is a decision support information used in decision making process. If only the data exists, there is no value and meaning to information in DSS environment. In the operational environment, its objective act as giving meaning not to new thing but to repetitive work. The DW metadata is maturer than the legacy metadata.

The quantity of information is expressed as a function of quantity of data and metadata. Also, the quality of information is expressed as a function of quality of data and metadata. The quality depends on the quantity. If much of metadata quantity exist, it is difficult to keep the metadata quality high. On the contrary, If a little of metadata quantity exist, it is easy to keep the metadata quality high. Thus, there exists a trade-off between the metadata quantity and the metadata quality. In the case of the metadata quantity, there are no restrictions on the content of metadata and size [53]. But, there is a trade-off metadata creation and maintenance, cost and resource. There is a temptation to try to put everything in the world in a dictionary, and there are several major pitfalls to this approach, and so the designer must choosy as to what goes into a dictionary [14]. Also whereas all metadata is not stored explicitly, much metadata is implicit [51]. Therefore it is necessary to have a core metadata. The core metadata stands on the basis of the standard. The standard is the starting point of the metadata quantity and quality.

The metadata standard for DW are designed for vendors' products to exchange metadata information. Efforts are under way by a consortium of vendors, known as the Metadata Council, to standardize metadata interchange between diverse vendor products within the DW arena [22]. The Metadata Council presented the entities and relationships which defined the metamodel for version 1.1 Metadata Interchange Standard (MIS) [39]. This supplies the interchange standard metamodel which consists of ASCII file format used to represent the metadata being exchanged [32]. Because this standard is being

developed slowly, additive items are well not included.

Also because this MIS metamodel has a small set of minimum required fields, in this paper the proposed metadata schema has added seven component items of DWAI covered through the DWE to the MIS metamodel. The added information for DWE is new requirements to MIS metamodel and is expressed into the proposed schema. This proposed metadata schema is called core metadata schema for DWE (Fig. 6). The core metadata schema consists of adding MIS metamodel in Formula (1.1) to architecture and infrastructure information in Formula (1.2). The final metadata schema in the metastore design/develop stage is derived as Formula (1.3).

$$\textit{MIS Metamodel} = \{\textit{Entity and Attribute Information}\} + \{\textit{Table Information}\} + \{\textit{Mapping Information}\} \quad (1.1)$$

$$\textit{Core Metadata Schema} = \{\textit{MIS Metamodel Information}\} + \{\textit{DWE Information}\} \quad (1.2)$$

$$\textit{Final Metadata Schema} = \{\textit{Core Metadata Schema Information}\} + \{\textit{Domain-specific Metadata Information}\} \quad (1.3)$$

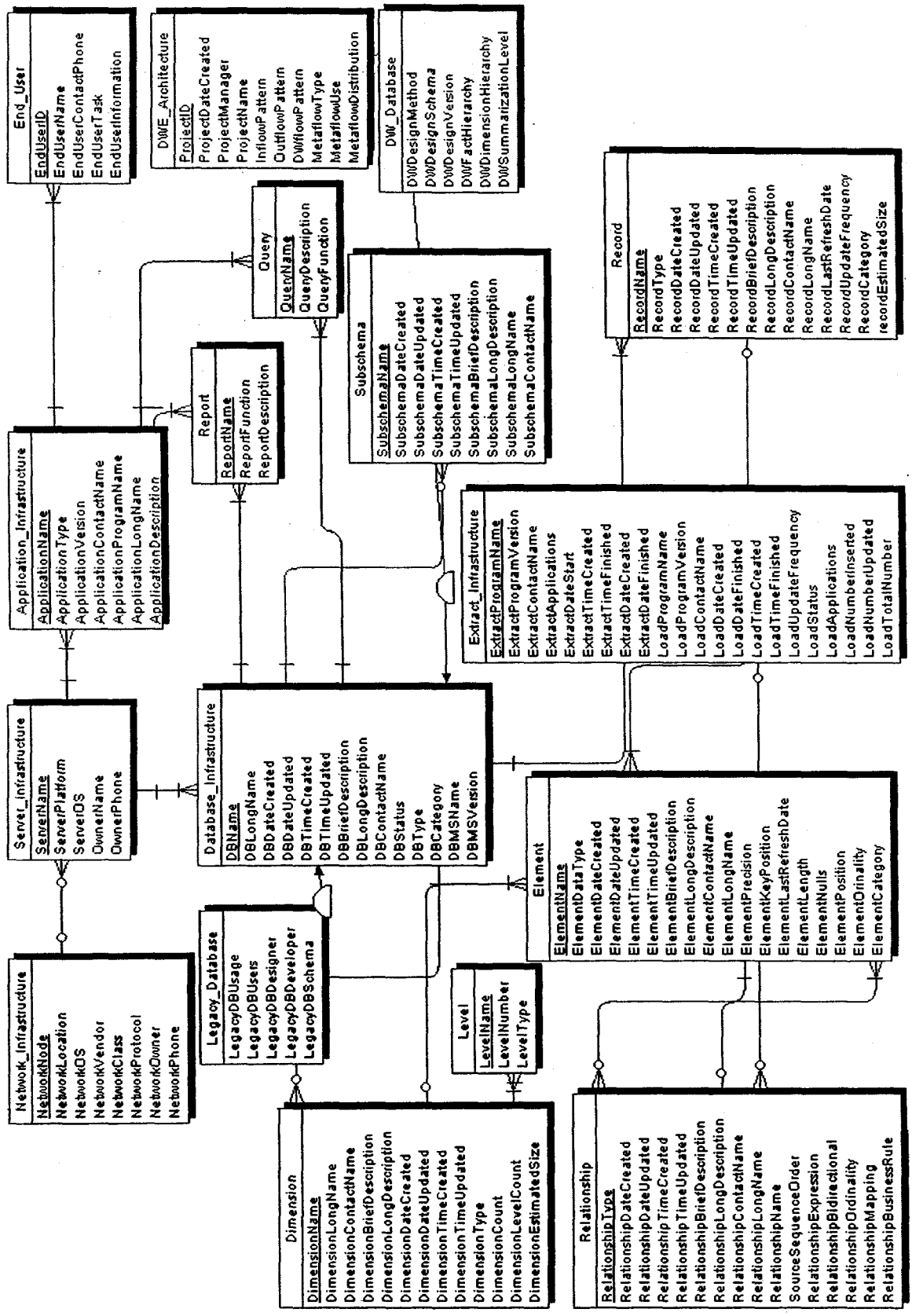

Fig. 6. Core metadata schema for DWE

What makes the whole issue complicated is that there is no single type of metadata and so, this has led to considerable marketplace confusion as vendors try to enhance their position by touting their products' metadata capabilities [24]. Table 7 compares with the some metadata schema in respect to criteria in the warehouse environment. Research on metadata classification is divided into by location [18, 25], by users [20, 49, 50, 53, 57], and by function [12, 24, 42], and so on. But, because DW is defined as architecture, metadata classification is evaluated and added by architecture as DWE such as in this paper.

When warehouse data is replicated to remote locations, corresponding metadata must also be replicated [40]. This research will proceed to have both the technical metadata and business metadata, and simultaneously to have the state that the metadata is distributed. This encases the DWE processing. But rather the business metadata than the technical is focused on because its environment is devised for decision support and is mainly for information hounds.

All of the prior researchers don't cover the entire DWE. In this Table, the proposed schema deals with the all process of DWE by inserting seven component of the environment. Also users' new requirements are able to include the schema iteratively. Most of schema is represented by entity-relationship model (ERM) techniques. Inmon, Zachman, and Geiger [28] and Barquin and Edelstein [2] simply proposed ER schema without the metadata development in implementing DW and Burton [7] proposed the partial metadata development after DW development. The posterior- development does not detailedly capture and manage the metadata, and failed at metadata management [26].

Metastore application system of searching and browsing metadata can be developed in accordance with application development stage in metastore development phase because metadata entry is varied as new requirements through the users' involvement. Future studies must concentrate on the metastore system architecture (Fig. 7) and system implementation applied into real project case. The system architecture may include modules having methodology supporting, metastore database, and application system.

Table 7 Comparison with the some metadata schema in the warehouse environment

Criteria	Inmon, Zachman, And Geiger [28]	Barquin and Edelstein [2]	Burton [7]	Proposed Schema
Classification	Mixed Format (Location, Users, and Function)	Function	None	DW Components
Framework	Zachman Framework	None	None	DW Environment
Elements	Motivation Entities Activities Location People Times	Query Processing Build-Time Manage	Table Field Source Usage	MIS Metamodel DWE Seven Components
Field Requirement Intervention	Including	None Fixed Metamodel	All Flexible	Mostly Reflection Flexible
Modeling Method	ERM	ERM	ERM	ERM
Standard Consideration	A Little Including	None	None	Including All Part of Standard
Development Methodology	None	None	Partially Insufficient; Not Detailed	Fully
DW Developing Relation	None	None	After DW implemented	Simultaneously Developing and Integrated with DW Development

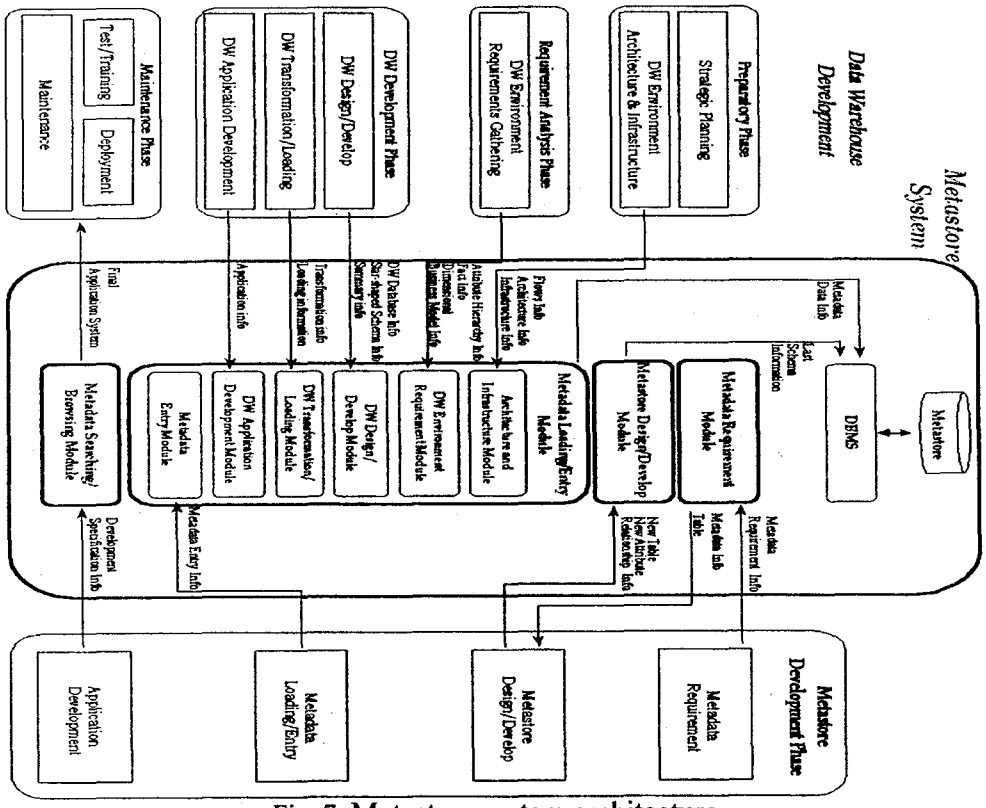

Fig. 7. Metastore system architecture

4. Conclusions

In today's information processing, the DW plays an important role in integrating enterprise information. Thus, we need to have a fresh look at DWE. It is impressive to understand that the DW is not an application, not an infrastructure, but an *architecture*. The DWA has seven components. Among its architecture, the metadata component is a key to successfully implement, manage, and use DW. In general, metadata definition is data about data. But, in the DWE, metadata definition is extended that metadata is information about enterprise. The metadata is composed of business metadata and technical metadata. Under this classification, metadata schema for DW is derived. Also its schema includes metadata interchange standard format. In this paper, the focus is to develop metadata, not to buy it, with and within the DWE development. The IDM differs from other development methodology in applying and integrating metadata development into this methodology. As the DW development interacts with metadata development, the metastore is developed in a concurrent fashion. The developed metastore is also a kind of metadata development support system.

The future work in this paper needs further research. First real-project including ODS and DM must be examined. If ODS and DM is inserted in DWE, the problems such as distributing metadata in metastore is taken into consideration. Metadata distribution is related to a distributed database technology. Second, in the case of DW via World Wide Web (WWW), the design of DW application is also considered. Together with it, DW metadata delivery via WWW is an important factor. The client application, web browser, easy-to-use it, lower cost are a competitive element [29]. Third, there is integration with enterprise resource planning (ERP). ERP system is designed to day-to-day OLTP system, not analytical system. But ERP source serves flexibility to use best of breed OLAP, analytical, and reporting tools [8]. The ERP system can be included in DWE architecture. Fourth, in integrated methodology, strategic planning stage and DWE gathering stage is added to delivery and refinement for part of metastore development. There could be much room for consideration into that takes workflow model in requirement stage. Fifth, another factor of considering the present research is object technology. The object technology has potential to make the architecture a much more flexible environment for data

warehousing and other needs for a true corporate metadata model [48]. Object-oriented software engineering (OOSE) approaches to data warehousing [16]. On the steps of the development methodology, object oriented techniques could be applied to architecture, analysis, modeling, and so on. Sixth, the metastore system must provide seamlessly connected access tool. This approach has many advantages [4].

References

[1] Atre, S., and Storer, P., October 1995, Data distribution and warehousing, *DBMS*, pp. 54-62.

[2] Barquin, and Edelstein, Panning and Designing the Data Warehouse, Prentice-Hall, inc., 1997.

[3] Bernstein et al., 1997, The Microsoft Repository, *23rd VLDB Conference.*

[4] Bischoff, J., et al., 1997, *Data Warehouse: Practical Advice from the Experts* (Prentice-Hall).

[5] Brooks P., October 1997, March of the data marts, *DBMS*, pp. 55-60.

[6] Brown, 1997, Preparing data for the Data Warehouse, *DCI' Data Warehouse World Conference.*

[7] Burton, et al., 1996, *Metadata: the Key to DW Design*, Univ. of Maryland, *White paper.*

[8] CSG, 1997, Price Warehouse, *Worldclass ERP '97 Conference.*

[9] Chaudhuri S., and Dayal U., March 1997, An overview of data warehousing and OLAP technology, SIGMOD Record, Vol. 26, No. 1, pp. 65-69.

[10] Demarest, M., 1994, Building the data mart, *DBMS*, v.7, n. 8, pp. 44 - 47.

[11] Denzer R., and Guttler R., 1996, Requirements of the meta information models for the environmental domain, Journal of computing and information, Vol. 2, No. 1, pp. 1328-1335.

[12] Devlin B., 1997, *Data Warehouse from Architecture to Implementation*, Addison-Wesley.

[13] Drewry M., Conover H., McCoy S., and Graves S. J., 1997, Metadata: quality vs. Quantity, 2nd IEEE Metadata Conference.

[14] Duyn J. V., 1982, *Developing a Data Dictionary System*, Prentice-Hall.

[15] Elkins, April 1998, Open OLAP, *DBMS*, pp. 35-45.

[16] Firestone, J. M., 1997, *Object-Oriented Data Warehousing*, EIS Inc., *White paper.*

[17] Golfarelli, Maio, and Rizzi, 1998, Conceptual design of data warehouses from E/R schemes, *Proceedings of the HICSS.*

[18] Griffin, Metadata: Capturing the heart of DW, ADT, 1996.

[19] Hackathorn R., February 1995, Data warehousing energizes your enterprise, *Datamation*, pp. 38-42.

[20] Hackney D., 1997, Metadata in 850 (or so) words, *Data Warehouse Forums.*

[21] Harding, December 1997, Olap meets the Web, *Application Development Trends*, pp. 70-73.

[22] Harjinder and Prakash, 1996, *The Official Guide to Data Warehousing*, Que.

[23] Hoven John van den, 1997, Data warehousing new name for the accessibility challenge, *Information Systems Management*, pp.70-72.

[24] Hurwitz J., July 1997, The evolution of metadata, DBMS, pp.12-14.

[25] Inmon, W. H., and Hackathorn, R. D., 1994, *Using the Data Warehouse* (John-Wiley).

[26] Inmon, W. H., 1995, Data warehouse success requires development automation, *Application Development Trends.*

[27] Inmon, W. H., Imhoff, C., and Battas, G., 1996, *Building the Operational Data Store* (John-Wiley).

[28] Inmon, Zachman, and Geiger, 1997, *Data Stores, Data Warehousing, and Zachman Framework*, McGraw-Hill.

[29] Ivy, 1997, Advances in DSS Metadata Delivery via the WWWeb, *CAUSE Conference.*

[30] Keen P. G. W., November 1997, Let's focus on action not information, *Computerworld.*

[31] Kelly, S., 1996, *Data Warehousing: the Route to Mass Customization* (John-Wiley).

[32] Kelly, 1997, *Data Warehousing in Action*, John-Wiley.

[33] Kimball R., October 1995, Is ER modeling hazardous to DSS, *DBMS*, pp. 17-28.

[34] Kimball R., The Data Warehouse Toolkit, John-Wiley, 1996.

[35] Kosar, D., Winter 1996, Architectural framework for building a warehouse, DB2, pp.10-15.

[36] Ladley, 1997, methodologies strategies for DW, *DCI's Data Warehouse World Conference.* pp. B21 - B21-9.

[37] Lauden, K. C., and Lauden, J. P, 1994, Management Information Systems: Organization and Technology, 3rd edition (Macmillan).

[38] McFadden, F. R., 1996, Data warehouse for EIS: some issues and impacts, *Proceedings of the 29th Annual HICSS*, 120-129.

[39] Metadata Council, 1997, Metadata Interchange Specification (MDIS), *White Paper.*

[40] Moriarty T., and Schmidt B., May 1997, Mining for metadata, *Database Programming and Design,* pp. 54-57.

[41] Moriarty T., and Mandracchia C, December 1996, Heart of the warehouse, Database Programming & Design, pp. 70-74.

[42] Moriarty, T., July 1997, What is metadata ?, Database Programming and Design, pp. 57-60.

[43] Orfali, O., Harkey, D., and Edwards, J., 1996, *The Essential Client/Server Survival Guide,* 2nd edition (John-Wiley).

[44] Orr K., 1995, Data warehouse technology, *White Paper.*

[45] Özsu M. T., and Valduriez P., 1991, *Principles of Distributed Database Systems*, Prentice-Hall.

[46] Poe V., July 1995, Data warehouse: architecture is not infrastructure, *Database Programming and Design.*

[47] Poe, V., 1995, *Building a Data Warehouse for Decision Support* (Prentice-Hall).

[48] Sachdeva S., December 1995, Guiding users through disparate data layers, *Application Development Trends*

[49] Sachdeva, Meta data architecture for data warehousing, *Data Management Review*, April 1998.

[50] Sherman R. P., August 1997, Metadata: the missing link, *DBMS*, pp. 73-82.

[51] Spiers, 1997, Metadata: Maximizing the Value of Information, *White Paper*

[52] Subramanian et al., 1996, Strategic planning for data warehousing in the public sector, HICSS.

[53] Sumpter, 1994, Whitepaper on data management, *White Paper.*

[54] Taylor D. A., 1995, *Business Engineering with Object Technology* (John-Wiley).

[55] Thomsen, June 1997, Dimensional hierachies and formulas, *Database Programming and Design,* pp. 60-63.

[56] Thomsen, May 1998, Darwin would be proud, *Database Programming and Design,* pp. 67-69.

[57] White C., February 1995, The key to a data warehouse, *Database Programming and Design,* pp. 23-25.

[58] Widom J., November 1995, Research problems in data warehousing, Proceedings of 4th International Conference on Information and Knowledge Management.

[59] Zornes, 1997, Capitalizing on data warehouse opportunities, DCI's Data Warehouse World Conference.