By Arun Sen and Atish P. Sinha

A Comparison of Data Warehousing Methodologies

Using a common set of attributes to determine which methodology to use in a particular data warehousing project.

DATA INTEGRATION TECHNOLOGIES have experienced explosive growth in the last few years, and data warehousing has played a major role in the integration process. A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data that supports managerial decision making [4]. Data warehousing has been cited as the highest-priority post-millennium project of more than half of IT executives. A large number of data warehousing methodologies and tools are available to support the growing market. However, with so many methodologies to choose from, a major concern for many firms is which one to employ in a given data warehousing project. In this article, we review and compare several prominent data warehousing methodologies based on a common set of attributes.

Online transaction processing (OLTP) systems are useful for addressing the operational data needs of a firm. However, they are not well suited for supporting decision-support queries or business questions that managers typically need to address. Such questions involve analytics including aggregation, drilldown, and slicing/dicing of data, which are best supported by online analytical processing (OLAP) systems.

Data warehouses support OLAP applications by storing and maintaining data in multidimensional format. Data in an OLAP warehouse is extracted and loaded from multiple OLTP data sources (including DB2, Oracle, IMS databases, and flat files) using Extract, Transfer, and Load (ETL) tools.

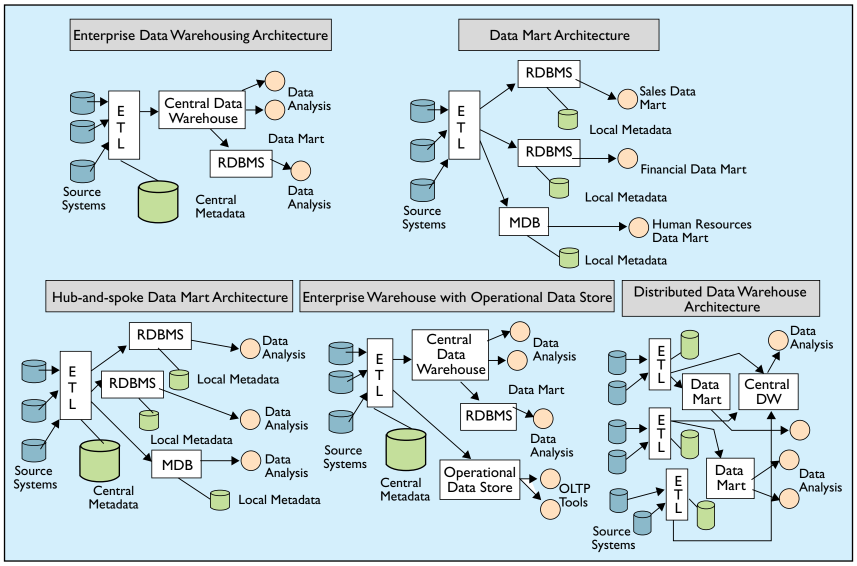
The warehouse is located in a presentation server. It can span enterprisewide data needs or can be a collection of “conforming” data marts [8]. Data marts (subsets of data warehouses) are conformed by following a standard set of attribute declarations called a data warehouse bus. The data warehouse uses a metadata repository to integrate all of its components. The metadata stores definitions of the source data, data models for target databases, and transformation rules that convert source data into target data.

The concepts of time variance and nonvolatility are essential for a data warehouse [4]. Inmon emphasized the importance of cross-functional slices of data drawn from multiple sources to support a diversity of needs [4]; the foundation of his subject-oriented design was an enterprise data model. Kimball introduced the notion of dimensional modeling [8], which addresses the gap between relational databases and multidimensional databases needed for OLAP tasks. These different definitions and concepts gave rise to an array of data warehousing methodologies and technologies, which we survey here and provide useful guidelines for future adopters.

Tasks in Data Warehousing Methodology

Data warehousing methodologies share a common set of tasks, including business requirements analysis, data design, architecture design, implementation, and deployment [4, 9].

For business requirements analysis, techniques such as interviews, brainstorming, and JAD sessions are used to elicit requirements. Business requirements analysis is used to elicit the business questions from the intended users of the data warehouse. Business questions are decision support or analytic questions that managers typically pose. After all the business questions are elicited, they are prioritized by asking the users to rate the questions, or by estimating the risk associated with the solutions needed for the questions. Next, a very high-level conceptual model (also known as the subject-area data model) of the solution for each of the business questions is created. The conceptual model serves as the blueprint for the data requirements of an organization.



Different types of data warehouse architectures.

The data design task includes data modeling and normalization. The two most popular data modeling techniques for data warehousing are Entity-Relational and Dimensional modeling. The Entity-Relational modeling follows the standard OLTP database design process, starting with a conceptual entity-relationship (ER) design, translating the ER schema into a relational schema, and then normalizing the relational schema.

A dimensional model is composed of a fact table and several dimension tables [8]. A fact table is a specialized relation with a multi-attribute key and contains attributes whose values are generally numeric and additive. A dimension table has a single attribute primary key (usually surrogate) that corresponds exactly to one of the attributes of the multi-attribute key of the fact table. The characteristic star-like structure of the physical representation of a dimensional model is called a *star join* schema, or simply a star schema. A dimensional model can be extended to a *snowflake schema*, by removing the low cardinality attributes in the dimensions and placing them in separate tables, which are linked back into the dimension table with artificial keys [9].

In the OLAP realm, decision-support queries may require significant aggregation and joining. To improve performance, denormalization is usually promoted in a data warehouse environment.

Architecture is a blueprint that allows communication, planning, maintenance, learning, and reuse. It includes different areas such as data design, technical design, and hardware and software infrastructure design. The architecture design philosophy has its origins in the schema design strategy of OLTP databases. Several strategies for schema design exist, such as top-down, bottom-up, inside-out, and mixed [1]. The data warehouse architecture design philosophies can be broadly classified into enterprisewide data warehouse design and data mart design [3]. The data mart

ALTHOUGH THE METHODOLOGIES USED BY THESE COMPANIES DIFFER IN DETAILS, THEY ALL FOCUS ON THE TECHNIQUES OF CAPTURING AND MODELING USER REQUIREMENTS IN A MEANINGFUL WAY.

Table 1. Comparison of core technology vendor-based data warehousing methodologies.

Attributes	NCR/Teradata Methodology	Oracle Methodology	IBM DB2 Methodology	Sybase Methodology	Microsoft SQL Server Methodology
Core Competency	Teradata DBMS (massively parallel DBMS)	Oracle DBMS	DB2 DBMS	Sybase DBMS	SQL Server DBMS
Requirements Modeling	Interview, JAD, Prioritization, templates, document analysis	Interview, Prioritization, subject areas	Interview, JAD	Interview	Interview document analysis
Data Modeling	ERD, relational schema	Dimensional model, Star schema	Dimensional model, Star schema	ERD, Star schema, Relational schema	Dimensional model, Star and Snowflake schemas
Support for Normalization/ Denormalization	Develops all relations as normalized, allows denormalization	Allows both	Allows both	More slanted toward denormalization	Allows both
Architecture Design Philosophy	Enterprise data warehouse	Data marts	Enterprise data warehouse and data marts	Data marts	Enterprise data warehouse and data marts
Implementation Strategy	Iterative	Dimensional Life Cycle	Iterative (prototyping)	Iterative (RAD)	Iterative
Metadata Management	Yes, uses a repository	Yes, uses Oracle Repository	Yes, uses a repository	Yes, uses a repository	Yes, uses Microsoft Repository
Query Design	Parallel query development	Allows parallel queries	Not reported	Not reported	Allows parallel queries
Scalability	Yes, to hundreds of Terabytes	Not reported	Yes	Not reported	Yes, to Terabytes
Change Management	Has post audit reviews, but not emphasized in the methodology	Not reported	Not reported	Has maintenance in the methodology	Not reported

Attributes	SAS Methodology	Informatica's Velocity Methodology	Computer Associates' Methodology	Visible Technologies' Methodology	Hyperion's STAR Methodology
Core Competency	Data analytics	Data analytics	Business Intelligence and Middleware	Business analysis software	Business analysis software and OLAP server
Requirements Modeling	Interview, JAD, document analysis	Business process inventory, JAD, subject areas	Interview, JAD, document analysis	Interview, JAD, prioritization, templates,document analysis	Analyze data sources and data sources
Data Modeling	ERD, Dimensional model, Relational schema	ERD, Dimensional model, Star schema	ERD, Dimensional model, Star schema	Warehouse model, ERD, Star schema	Dimensional model, Star schema
Support for Normalization/ Denormalization	Not reported	Not reported	Not reported	Allows both	Allows both
Architecture Design Philosophy	Enterprise data warehouse and data marts	Enterprise data warehouse with data marts	Enterprise data warehouse with data marts	Enterprise data warehouse with data marts	Enterprise data warehouse with data marts
Implementation Strategy	Iterative	Iterative spiral	Iterative (Piloting/ prototyping)	Iterative	Iterative
Metadata Management	Yes. Uses integrated metadata management	Yes. Uses an integrated metadata platform	Yes. Uses its own repository	Yes. Uses its own repository	Yes
Query Design	Depends on the DBMS to be used at the warehouse level	Allows parallelism	Not reported	Not reported	Allows parallelism via partitioning
Scalability	Yes	Yes	Yes	Yes	Yes
Change Management	Very little	Very little	Yes	Uses Visible tools	Not reported

Table 2. Comparison of infrastructure-based data warehousing methodologies.

design, espoused by Kimball [8], follows the mixed (top-down as well as bottom-up) strategy of data design. The goal is to create individual data marts in a bottom-up fashion, but in conformance with a skeleton schema known as the "data warehouse bus." The data warehouse for the entire organization is the union of those conformed data marts. The figure on the preceeding page depicts several variants of the basic architectural design types, including a hub-and-spoke architecture, enterprise warehouse with operational data store (real-time access support), and distributed enterprise data warehouse architecture [2].

Data warehouse implementation activities include data sourcing, data staging (ETL), and development of decision support-oriented end-user applications. These activities depend on two things—data quality management and metadata management [5, 7]. As data is gathered from multiple, heterogeneous OLTP sources, data quality management is a very important issue. A data warehouse generates much more metadata than a traditional DBMS. Data warehouse metadata includes definitions of conformed dimensions and conformed facts, data cleansing specifications, DBMS load scripts, data transform runtime logs, and other types of metadata [9]. Because of the size of metadata, every data warehouse should be equipped with some type of metadata management tool.

For data warehouse implementation strategy, Inmon [4] advises against the use of the classical Systems Development Life Cycle (SDLC), which is also known as the waterfall approach. He advocates the reverse of SDLC: instead of starting from requirements, data warehouse development should be driven by data. Data is first gathered, integrated, and tested. Next, programs are written against the data and the results of the programs are analyzed. Finally, the requirements are formulated. The approach is iterative in nature.

Kimball et al.'s business dimensional life-cycle approach "differs significantly from more traditional, data-driven requirements analysis" [9]. The focus is on analytic requirements that are elicited from business managers/executives to design dimensional data marts. The life-cycle approach starts with project planning and is followed by business requirements definition, dimensional modeling, architecture design, physical design, deployment, and other phases.

Attributes	SAP Methodology	PeopleSoft Methodology	CGEY Methodology	Corporate Information Designs Methodology	Creative Data Methodology
Core Competency	ERP	ERP	General business consulting	IT consulting	Business Intelligence consulting
Requirements Modeling	Interview templates	Interview	Follows varied approach (SAP, Microsoft, Oracle and Peoplesoft)	Subject areas, data granularities, etc.	Interviews, JAD, document analysis
Data Modeling	Dimensional model, Extended star schema	Predefined data warehouse model, Dimensional model, Star schema	Dimensional model, Star schema	ERD/Object model, Relational schema	Dimensional Model, Star schema
Support for Normalization/ Denormalization	Allows denormalization	Not reported	Follows multiple strategies	Not reported	Not reported
Architecture Design Philosophy	Enterprise data warehouse and data marts	Enterprise data warehouse and data marts	Data marts	Enterprise data warehouse and data marts	Enterprise data warehouse and data marts
Implementation Strategy	Iterative (prototyping)	SDLC	Follows steps used by the type chosen at the requirements level	SDLC (waterfall), Iterative (spiral)	Iterative (RAD)
Metadata Management	Integrated meta data repository	Yes	Not reported	Yes	Not reported
Query Design	Allows ad hoc queries	Allows ad hoc queries	Allows ad hoc queries	Not reported	Not reported
Scalability	Yes	Integrated and scalable open architecture	Yes	Yes	Yes
Change Management	Different modeling methods for tracking history	Allows impact analysis	Not reported	Not reported	Not reported

Table 3. Comparison of information modeling-based data warehousing methodologies.

For enterprisewide data warehouse development, it is impractical to determine all the business requirements a priori, so the SDLC (waterfall) approach is not viable. To elicit the requirements, an iterative (spiral) approach such as prototyping is usually adopted. Individual data marts, on the other hand, are more amenable to a phased development approach such as business dimensional life cycle because they focus on business processes, which are much smaller in scope and complexity than the requirements for an enterprisewide warehouse.

The deployment task focuses on solution integration, data warehouse tuning, and data warehouse maintenance. Although solution integration and data warehouse tuning are essential, maintenance is cited as one of the leading causes of data warehouse failures. Warehouses fail because they do not meet the needs of the business, or are too difficult/expensive to change with the evolving needs of the business. Due to increased end-user enhancements, repeated schema changes, and other factors, a data warehouse usually goes through several versions.

Comparing Data Warehousing Methodologies

We analyzed 15 different data warehousing methodologies, which we believe are fairly representative of the range of available methodologies (see Tables 1–3). The sources of those methodologies can be classified into three broad categories: core-technology vendors, infrastructure vendors, and information modeling companies. Based on the data warehousing tasks described earlier, we present a set of attributes that capture the essential features of any data warehousing methodology.

Core Competency Attribute. The first attribute we consider is the core competency of the companies, whose methodologies could have different emphases depending upon the segment they are in. The core-technology vendors are those companies that sell database engines. These vendors use data warehousing schemes that take advantage of the nuances of their database engines. The methodologies we review include NCR's Teradata-based methodology, Oracle's methodology, IBM's DB2-based methodology, Sybase's methodology, and Microsoft's SQL Server-based methodology.

The second category, infrastructure vendors, includes those companies that are in the data warehouse infrastructure business. An infrastructure tool in the data warehouse realm could be a mechanism to manage metadata using repositories, to help extract, transfer, and load data into the data warehouse, or to help create end-user solutions. The infrastructure tools typically work with a variety of database engines.

Change management is an important issue to consider in selecting a data warehousing methodology. Surprisingly, very few vendors incorporate change management in their methodologies.

The methodologies proposed in this category, therefore, are DBMS-independent. Such methodologies include SAS's methodology, Informatica's methodology, Computer Associates' Platinum methodology, Visible Technologies' methodology, and Hyperion's methodology.

The third category, information modeling vendors, includes ERP vendors (SAP and PeopleSoft), a general business consulting company (Cap Gemini Ernst Young), and two IT/data-warehouse consulting companies (Corporate Information Designs and Creative Data).

We include ERP vendors because data warehousing can leverage the investment made in ERP systems. Data warehousing is a technology service for most consulting companies, including general ones like Cap Gemini Ernst Young (CGEY) or specific ones like Corporate Information Designs and Creative Data. We group the ERP and consulting companies into one category because of the similarities in their objectives. Although the methodologies used by these companies differ in details, they all focus on the techniques of capturing and modeling user requirements in a meaningful way. Therefore, the core competency of this category is information modeling of the clients' needs.

Requirements Modeling Attribute. This attribute focuses on techniques of capturing business requirements and modeling them. For building a data warehouse, understanding and representing user requirements accurately is very important. Data warehouse methodologies, therefore, put a lot of emphasis on capturing business requirements and developing information models based on those requirements.

Various types of requirements elicitation strategies are used in practice, ranging from standard systems development life-cycle techniques such as interviews and observations to JAD sessions. As this elicitation process is fairly unstructured, several methodologies use streamlining tricks. Examples include NCR/Teradata's elicitation and prioritization of business questions, Oracle and Informatica's creation of subject areas, and NCR/Teradata and Sybase's template-directed elicitation.

Data Modeling Attribute. This attribute focuses on data modeling techniques that the methodologies use to develop logical and physical models. Once the requirements are captured, an information model (also called a warehouse model) is created based on those requirements. The model is logically represented in the form of an ERD, a dimensional model, or some other type of conceptual model (such as an object model). The logical model is then translated into a relational schema, star schema, or snowflake schema during physical design. NCR/Teradata, SAS, and Informatica provide examples of methodologies that map an ERD into a set of normalized relations. In the Sybase methodology, a conceptual ERD is first translated into a dimensional model. Other vendors, including IBM, Oracle, SAP, and Hyperion, use the dimensional model for logical design and the star schema for physical design.

Support for Normalization/Denormalization Attribute. The normalization/denormalization process is an important part of a data warehousing methodology. To support OLAP queries, relational databases require frequent table joins, which can be very costly. To improve query performance, a methodology must support denormalization. We found that all DBMS vendors explicitly support the denormalization activity. Other vendors listed in Tables 2 and 3 do not report this capability much, possibly due to the fact that they depend on the DBMS to be used.

Architecture Design Philosophy Attribute. A number of strategies are available for designing a data warehouse architecture, ranging from enterprisewide data warehouse design to data mart design. The organization needs to determine which approach will be the most suitable before adopting a methodology.

Implementation Strategy Attribute. Depending on the methodology, the implementation strategy could vary between an SDLC-type approach and a RAD-type approach. Within the RAD category, most vendors have adopted the iterative prototyping approach.

Metadata Management Attribute. Almost all vendors focus on metadata management, a very important aspect of data warehousing. Some DBMS vendors (Oracle, Teradata, IBM, Sybase, and Microsoft) and some infrastructure vendors (Informatica, Computer Associates, and Visible Technology) have an edge because they have their own repository systems to manage metadata.

Query Design Attribute. Large data warehouse tables take a long time to process, especially if they must be joined with others. Because query performance is an important issue, some vendors place a lot of emphasis on how queries are designed and processed. Some DBMS vendors allow parallel query generation and execution. This is a predominant feature in NCR's Teradata DBMS and is therefore included in its methodology. Teradata is a truly parallel DBMS, providing strong support for parallel query processing. Vendors like Microsoft and Oracle allow parallel queries, but process them in a conventional fashion. Other vendors listed in tables 2 and 3 depend on the DBMS they use.

Scalability Attribute. Although all methodologies support scalability, note that scalability is highly dependent on the type of DBMS being used. In Teradata, for example, scalability can be achieved by adding more disk space, while in others, increasing the size may require considerable effort. However, the cost of the proprietary hardware, specialized technical support, and specialized data loading utilities in Teradata result in higher overhead and development costs than DB2, Oracle, Sybase, or SQL Server. Teradata does not economically scale down below a terabyte. Organizations should consider this issue before selecting a data warehousing methodology.

Change Management Attribute. Various changes affect the data warehouse [6]. For a large number of enterprises in today's economy, acquisition is a normal strategy for growth. An acquisition or a merger could have a major impact. For a data warehouse project, it could imply rescoping of warehouse development, replanning priorities, redefining business objectives, and other related activities. Company divestiture is also another source of changes for any enterprise, but has a less severe impact on a data warehouse. Newer technologies could also affect the way an e-commerce site is set up and introduce changes. With advances in portal technology, expansion of bandwidth, and efforts to standardize models, firms could be reconfiguring their Web sites, thereby initiating a lot of changes.

Changes in the physical world also affect the data warehouse. For example, customers frequently change their addresses. Sales regions get reconfigured. Products get assigned to new categories. Sometimes it is important to capture those changes in the warehouse for future analyses. Changes in process are part of the natural evolution of any enterprise. An intelligent enterprise should be able to manage and evaluate its business processes. An example of a process change is introducing new data cleansing routines, or adding new data sources, which would necessitate managing additional load scripts, load map priorities, and backup scripts. As the data warehouse implementation effort progresses, additional user requests and enhancements will inevitably arise. Those changes need to be handled, recorded, and evaluated. With OLAP front-end tools, there could be various changes to the front-end interface, such as addition of new front-end objects initially not available, changes in object definitions, and deletion of obsolete front-end objects.

Change management is an important issue to consider in selecting a data warehousing methodology. Surprisingly, very few vendors incorporate change management in their methodologies. When they do, it is usually masked as maintenance. The Visible Technologies methodology strongly focuses on change management and has tools to support this process.

Conclusion

Data warehousing methodologies are rapidly evolving but vary widely because the field of data warehousing is not very mature. None of the methodologies reviewed in this article has achieved the status of a widely recognized standard as yet. As the industry matures, there could be a convergence of the methodologies, similar to what happened with database design methodologies. It is apparent that the core vendor-based methodologies are appropriate for those organizations that understand their business issues clearly and can create information models. Otherwise, the organizations should adopt the information-modeling based methodologies. If the focus is on the infrastructure of the data warehouse such as metadata or cube design, it is advisable to use the infrastructure-based methodologies. C

References

1. Batini, C., Ceri, S., and Navathe, S.K. *Conceptual Database Design: An Enity-Relationship Approach*. Benjamin/Cummings, Redwood City, CA, 1992.
2. *DCI Seminar Workbook—Strategies and Tools for Successful Data Warehouses*. DCI, Andover, MA, 1999; www.dciexpo.com.
3. Hackney, D. *Understanding and Implementing Successful Data Marts*. Addison-Wesley, Reading, MA, 1997.
4. Inmon, W.H. *Building the Data Warehouse, 3rd edition*. Wiley, New York, 2002.
5. Inmon, W. Metadata in the data warehouse: A statement of vision. White Paper, Tech Topic 10, Pine Cone Systems, Colorado, 1997; www.inmoncif.com/library/whiteprs/techtopic/tt10.pdf.
6. Inmon, W. and Meers, D.P. The dilemma of change: Managing changes over time in the data warehouse/DSS environment. White Paper, 2001; www.kalido.com.
7. Inmon, W. Metadata in the data warehouse, White Paper, 2000; www.inmoncif.com/library/whiteprs/earlywp/ttmeta.pdf.
8. Kimball, R. and Ross, M. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd edition*, Wiley, New York, 2002.
9. Kimball, R., Reeves, L., Ross, M., and Thronthwaite, W. *The Data Warehouse Lifecycle Toolkit*. Wiley, New York, 1998.

Arun Sen (Asen@cgsb.tamu.edu) is a full professor and Mays Fellow in the Department of Information and Operations Management in Mays Business School at Texas A&M University.
Atish P. Sinha (sinha@uwm.edu) is an associate professor in the School of Business Administration at the University of Wisconsin-Milwaukee.