Data Warehouse Architecture

A Blueprint for Success

This paper describes methods for developing and documenting data warehouse architecture for strategic information management.

The key to success in scaleable data warehouse development and the single factor that contributes most to data warehousing success is Data Warehouse Architecture.

Table of Contents

Why Do You Need a Data Warehouse Architecture?

To date, too much emphasis has been placed on the raw technology which embodies the concept of Data Warehousing and not enough on the underlying and concomitant strategy, planning, business processes, and services which develop, use, and maintain the Data Warehouse technology. Experience has shown us that in projects where the technology has been perceived as a failure, the problem does not usually lie with the technology itself, but rather with the way in which the technology was applied. -- IBM Technology Guide, "Getting Started with Data Warehousing"

The most successful Data Warehouses will take an enterprise view of the requirements, perform a quick analysis of the information need to support all requirements, then work deeply in one area to build the first increment of the Data Warehouse. -- Doug Ebel, NCR Business Solutions Architect, "Data Warehousing -- Start to Start"

At the heart of any Data Warehouse is your data modeled to represent your business. As the warehouse is used to change your business, so must the warehouse change to continue to reflect an accurate model of the business. -- NCR White Paper, "The Heart of a Data Warehouse"

...Information Technology (IT) professionals should think of their [data warehouse] challenge not as data "scrubbing," but data reengineering -- a methodology and technology for lexically analyzing, reconditioning, enriching, and integrating data from multiple sources. -- George Burch, Chief Technology Officer, Vality Technology, "The Data Doctor Column," Data Warehouse Resources

Data Warehouse development, unlike transactional system development is a highly iterative process… Users' needs must be weighed… This means prioritization, more development projects, closer coordination among project teams and establishment of techniques and standards… Above all else, consistency of design and consistence across the data is absolutely essential. -- Vince Desio, Senior Consultant, Enterprise Solutions, Inc. "The Impact of Data Warehouse on MIS"

As indicated in the preceding extracts, successful Data Warehousing is not "business as usual."

In a 1998 in-depth analysis report, the META Group described the following emerging issues as most significantly altering the data warehousing approach taken by many organizations:

- **Inadequate Planning Assumptions for Large Data Warehouse Initiatives.** Once into the implementation stage, too many designers and project leaders for corporate data warehouse initiatives discover that their assumptions about the impact of organizational, political, and cultural dynamics — as well as their assessment of technology and skills shortfalls — were naïve and, in fact, major obstacles.
- **Technology Limitations.** A common stumbling block for corporate data warehousing is that even after the data is scrubbed, rationalized, and loaded, the final database is too large to fit on any known server. In addition, even the most focused data mart solution

can grind to a halt as new sources of internal and external data identified as critical to its business intelligence analysis deluge the system. Moreover, the processing power required to support access by the expanding user constituencies (encompassing employees, business partners, and ultimately customers) is beyond what is currently available. Massively parallel processing (MPP) systems and even clusters of inexpensive symmetric multiprocessing (SMP) systems often turn out to be white elephants, because these massive, centralized databases are too difficult to design, manage, and tune.

- **Data Mart Proliferation.** Business users are unable to tolerate the long lead times typical of even the best planned and executed data warehouse. As a result, many organizations focus efforts on data marts rather than corporate/enterprise-wide data warehousing. The uncoordinated architecture(s) that result often compromise the corporate IT infrastructure plans and result in future major diseconomies.
- **Exploding User Constituencies.** Organizations continue to state ambitious plans to increase the number of online analytical users for their data warehouses. Yet, year-to-year measurements show that these same organizations consistently fail to deploy at such magnitudes. Meanwhile, the projected number of users is again restated each year for presumed emerging constituencies — including direct consumer access to certain data warehouses.

The META Group report further describes the following obstacles to Data Warehouse success:

- Data Quality (41%)
- Managing End-User Expectations (31%),
- Legacy Data Transformation (28%),
- Business Rule Analysis (28%),
- Business Data Modeling (25%), and
- Managing Management Expectations (23%).

A Data Warehouse Architecture is the single solution to overcoming these obstacles, addressing these issues, and successfully developing Data Warehouses.

What Is a Data Warehouse Architecture?

Linking an enterprise's strategic information requirements (business architecture) with its information architecture, service component (processes, systems, and applications) architecture, and technical architecture results in an "*Enterprise Architecture*" (see author's white paper, "Enterprise Architecture Engineering," for more detail). A subset of the Enterprise Architecture is the Data Warehouse Architecture.

A well-documented architecture (for the enterprise and its data warehouse) is a logical organization of information pertaining to the following corporate-level, enterprise-wide elements:

- Strategic goals, objectives, and strategies
- Business rules and metrics
- Information requirements
- Processes, systems and applications

- Relationships between architecture elements
- Technology infrastructure

Data Warehouse Architecture also establishes guidelines, standards, and operational services that define the enterprise's operational system environment. When an enterprise's architecture is so documented, it can be used to accomplish the following:

- Facilitate change management by linking strategic requirements to systems (including the data warehouse and data marts) that support them and by linking the business model to application designs, including data warehouse designs
- Enable strategic information to be consistently and accurately derived from operational (and external) data
- Promote data sharing, thus reducing data redundancy and maintenance costs
- Improve productivity through component development, management, and reuse

The architecture and design of an enterprise's data warehouse should reflect the performance measurement and business requirements of the enterprise. Its **data model, structure, components,** and **metadata** should all be based upon internal information requirements -- not specific technologies.

BLUEPRINT FOR A DATA WAREHOUSE

"Engineering" a data warehouse is like engineering a physical warehouse. Both involve a rigorous development cycle and require the right tools.

A building is constructed using architectural diagrams (blueprints) that clearly depict the building's infrastructure (structural elements, walls, electrical wiring, plumbing, etc.). The best data warehouses are built from architectural models of enterprise infrastructure (policies, goals, measures, critical success factors, etc.).

Blueprints are also used to enlarge a building or make any significant modifications. Without a diagram of the infrastructure, such changes are quite difficult and very costly and can even be dangerous. It is the same with data warehouses. First update the enterprise's data warehouse architecture model so that it reflects changes (e.g., new performance measures, product lines, or services) and then modify the data warehouse to support the changed enterprise.

Data warehouse engineering is easier and less costly when based upon an accurate architectural model of the enterprise. Further, a data warehouse is easier to use and consistently produces desired outcomes when decision-makers have access to an enterprise architecture (metadata) that accurately reflects enterprise infrastructure.

Data Warehouse Data Model

A **data model** documents the data elements whose values at any point in time are necessary to tell data warehouse users how well their enterprise is performing. The data warehouse model provides a clear and unambiguous definition of every key data entity, describing the way each is used, as well as defining derivation formulas, aggregation categories, and refreshment time periods. The data warehouse model, linked with the enterprise information architecture, becomes both requirement documentation and a source for communicating the contents of the data warehouse to its users and developers. Issues that must be addressed in the data model include what legacy data will be used to populate the data warehouse, how data will be moved from legacy

environments to the data warehouse, and how the legacy data will be integrated or transformed to ensure data quality and integrity in the data warehouse. *The two most important issues for any data warehouse are data quality and data access*.

Data Warehouse Metadata

Metadata, or data about data, is the nerve center of a data warehouse and is essential. Metadata is essential to all levels of the data warehouse, but exists and functions in a different dimension from other warehouse data. Metadata used to manage and control data warehouse creation and maintenance resides outside the data warehouse, often in a digital repository. Metadata for data warehouse users is part of the data warehouse itself and is available to control access and analysis of the data warehouse. To a data warehouse user, metadata is like a "card catalog" to the subjects contained in the data warehouse. The two types of data warehouse metadata are called structural and access.

> **Structural metadata** is used for creation and maintenance of the data warehouse. It fully describes data warehouse structure and content. The basic building block of structural metadata is the data warehouse model that describes its data entities, their characteristics, and how they are related to one another. The way potential data warehouse users currently use, or intend to use, enterprise measures provides insight into how to best serve them from the data warehouse; i.e., what data entities to include and how to aggregate detailed data entities. The data warehouse model provides a means of documenting and identifying structural metadata. This includes both strategic and operational uses of enterprise measures, as well as multi-dimensional summarization. Structural metadata also includes performance metrics for programs and queries so that users and developers know how long programs and queries should run. Data warehouse performance tuning also uses these metrics.
>
> **Access metadata** is the dynamic link between the data warehouse and end-user applications. It generally contains the enterprise measures supported by the data warehouse and a dictionary of standard terms including user-defined custom names and aliases. Access metadata also includes the location and description of data warehouse servers, databases, tables, detailed data, and summaries along with descriptions of original data sources and transformations. Access metadata provides rules for drill up, drill down and views across enterprise dimensions and subject hierarchies like products, markets, and customers. Access metadata also allows rules for user-defined custom calculations and queries to be included. In addition, access metadata contains individual, work group, and enterprise security for viewing, changing, and distributing custom calculations, summaries, or other analyses.

Data Warehouse Components

The data warehouse architecture also contains descriptions data warehouse components**: current detail, summarized data,** and **archives** as well as **systems of record** and **integration/transformation programs**.

> The heart of a data warehouse is its **current detail**. It is the place where the bulk of data resides. Current detail comes directly from operational systems and may be stored as raw data or as an aggregation of raw data. Current detail, organized by subject area, represents the entire enterprise, rather than a given application. Current detail is the lowest level of data granularity in the data warehouse. Every data entity in current detail is a snapshot, at a moment in time, representing the instance when the data are accurate. Current detail is typically maintained for two to five years, but some enterprises may require detail

data for significantly longer periods. When initially implemented, a data warehouse may include current detail more than two years old, but the often questionable quality of older data must be considered and measures taken to ensure its validity. Current detail refreshment occurs as frequently as necessary to support enterprise requirements.

Lightly summarized data are the hallmark of a data warehouse. All enterprise elements (department, region, function, etc.) do not have the same information requirements, so effective data warehouse design provides for customized, lightly summarized data for every enterprise element (see Data Mart, below). An enterprise element may have access to both detailed and summarized data, but typically much less than the total stored in current detail.

Highly summarized data are primarily for enterprise executives. Highly summarized data can come from either the lightly summarized data used by enterprise elements or from current detail. Data volume at this level is much less than other levels and represents an eclectic collection supporting a wide variety of needs and interests. In addition to access to highly summarized data, executives also should have the capability of accessing increasing levels of detail through a "drill down" process.

Data warehouse **archives** contain old data (normally over two years old) of significant, continuing interest and value to the enterprise. There is usually a massive amount of data stored in the data warehouse archives that has a low incidence of access. Archive data are most often used for forecasting and trend analysis. Although archive data may be stored with the same level of granularity as current detail, it is more likely that archive data are aggregated as they are archived. Archives include not only old data (in raw or summarized form); they also include the *metadata* that describes the old data's characteristics.

A **system of record** is the source of the best or "rightest" data that feed the data warehouse. The "rightest" data are those which are most timely, complete, accurate, and have the best structural conformance to the data warehouse. Often the "rightest" data are closest to the source of entry into the production environment. In other cases, a system of record may be one containing already summarized data. Often, “rightest” data is created from diverse sources through a reconciliation process.

The components that link operational systems with the data warehouse are the **integration/transformation programs**. Even the "rightest" operational data cannot usually be copied, as is, into a data warehouse. Raw operational data are virtually unintelligible to most end users. Additionally, operational data seldom conform to the logical, subject-oriented structure of a data warehouse. Further, different operational systems represent data differently, use different codes for the same thing, squeeze multiple pieces of information into one field, and more. Operational data can also come from many different physical sources: old mainframe files, non-relational databases, indexed flat files, even proprietary tape and card-based systems. Thus operational data must be cleaned up, edited, and reformatted before being loaded into a data warehouse.

As operational data items pass from their systems of record to a data warehouse, integration and transformation programs convert them from application-specific data into enterprise data. These integration and transformation programs perform functions such as:

- Reformatting, recalculating, or modifying key structures and other data elements.
- Adding time elements
- Identifying default values

- Supplying logic to choose between multiple data sources
- Summarizing, tallying, and merging data from multiple sources
- Reconciling data from multiple sources

When either operational or data warehouse environments change, integration and transformation programs must be modified to reflect that change.

Data Warehouse Structure

A data warehouse may have any of several **structures**. The structure that best meets the data warehouse needs of an enterprise is fully dependent upon the enterprise business, data, and access requirements. The basic data warehouse structures are:

Physical Data Warehouse - physical database in which all the data for the data warehouse are stored, along with metadata and processing logic for scrubbing, organizing, packaging and processing the detail data.

Logical Data Warehouse - also contains metadata including enterprise rules and processing logic for scrubbing, organizing, packaging and processing the data, but does not contain actual data. Instead it contains the information necessary to access the data wherever they reside. This structure is possible *only* when operational systems exactly reflect the enterprise data architecture and system capacities can support both operational and management functions.

Data Mart - subset of an enterprise-wide data warehouse. Typically it supports an enterprise element (department, region, function, etc.). The organization of data in a data mart reflects the needs of the enterprise element it supports, and may be different from the organization of the enterprise data warehouse. Specific data elements may be stored redundantly in both the data mart and the data warehouse. As part of an iterative data warehouse development process, an enterprise builds a series of physical data marts over time and links them via an enterprise-wide logical data warehouse or feeds them from a single physical warehouse.

Both within the Data Warehouse as a whole and within the individual Data Marts, different groups of users have needs for differing slices of data. For example, users at a branch generally need the “horizontal slice” of data that pertains to their branch (i.e. they need all the data elements - tables and columns - but only the rows pertaining to their branch). Other users need “vertical slices” or a combination of horizontal and vertical slices.

Data Warehouse Engineering

"Data Warehouse" and "Data Warehouse Architecture” are relatively new terms that describe the methods and concepts that have used for almost twenty years to develop and implement Executive Information Systems (EIS), Decision Support Systems (DSS), and Management Information Systems (MIS). These years of practical experience have been distilled as best practices that make up our approach to data warehouse development.

The key to success in scaleable data warehouse development is using an iterative approach that includes active participation of potential data warehouse users. Designing and developing a data warehouse involves five key activities:

(1) Establish sponsorship;

(2))Identify enterprise needs;

(3) Develop data warehouse architecture;

(4) Apply appropriate technology; and

(5) Implement the data warehouse.

A detailed discussion of our approach is in the author's white paper "*Data Warehouse Engineering*"

Summary

Having a Data Warehouse Architecture as the blueprint for data warehouse engineering will help you deliver effective strategic information that exactly meets the needs of your enterprise -- public or private, large or small -- to the right people, in the right place, at the right time, in the right format.

For more information please contact:

Visible Systems Corporation
711 Atlantic Avenue
Boston, MA
contact@visiblesystemscorp.com

Alan Perkins has been a Systems Analyst on the White House staff, Director of the US Army Data Processing School in Germany, Vice President of R&D for a virtual corporation, Vice President of Consulting for Visible Systems Corporation and General Manager of a high-tech consulting firm. He has provided information and enterprise management consulting to numerous companies, associations and government agencies.

Mr. Perkins specializes in Enterprise Architecture Engineering. He helps clients quickly engineer enterprise architectures that are actionable and adaptable. His approach results in architectures that enable and facilitate enterprise initiatives such as Corporate Portals, Enterprise Data Warehouses, Enterprise Application Integration, Software Component Engineering, etc.

The following are papers are available at www.visiblesystemscorp.com:

"Enterprise Architecture Engineering"

"Enterprise Architecture Engineering Critical Success Factors"

"*High-Performance* Enterprise Architecture Engineering – Implementing the Zachman Framework for Enterprise Architecture"

"Enterprise Change Management – An Architected Approach"

"Getting Your Acts Together – An Architected Solution for Government Transformation"

"A Strategic Approach to Data Warehouse Engineering"

"Data Warehouse Architecture – A Blueprint For Success"

"Critical Success Factors for Data Warehouse Engineering"

"How to Succeed in the 21st Century – Critical Information Management Success Factors"

"XML Metadata Management – Controlling XML Chaos"

"Business Rules Are Meta-Data"

"Enterprise Application Modernization – Solving IT's Biggest Problem"

"Strategic Enterprise Application Integration"

"e-Engineering – A Unified Method"

"Enterprise Portal Engineering"

"Quality Software [Component] Engineering"

"Software Engineering Process Improvement"