**Enterprise Data Architecture: Why, What and How**

The goal of data architecture is to introduce structure, control and consistency to the fragmented data landscape found in most large enterprises. The results can be cost savings and improvements in business agility.

Most large enterprises live in a data landscape that is highly fragmented and contains significant redundancy. Having evolved over time as a result of mergers and acquisitions, custom application development and the use of packaged applications, the technology supporting this data infrastructure includes multiple database management systems (DBMSs) and various tools for data movement and integration. In addition, a large number of point-to-point interfaces usually ties these fragments together. The end result, when represented graphically, is the classic "spaghetti diagram" of an environment, which is often so complex that no one in the organization can understand it.

Enterprise data architecture is the process of managing data across the enterprise with the objective of resolving — or at least minimizing — these issues. By seeking opportunities for simplification, removing unnecessary data movement and fostering semantic, design and technology standards, enterprises can achieve cost savings and establish a more-flexible foundation to support strategic business initiatives.

**The Need for Enterprise Data Architecture (Why)**

For many years, the view was held that data diversity could be resolved by all applications sharing a common database. However, for most enterprises, this goal was elusive. It is hard to build a compelling business case to rewrite applications simply to support a new database design. Many of the projects that were approved did not fulfill their objectives, because they were too big, took too long, cost too much and involved excessive risk. At the same time, packages that offered attractive alternatives for specific problem domains were underpinned by their own database design. With the rise in popularity of enterprise applications, a broader range of needs was addressed. However,
no single vendor can supply all applications for a midsize or large enterprise, so data fragmentation and redundancy persisted.

The trend to buy applications or components rather than build them continues, which means that, accompanied by the endurance of many legacy applications and the need to interact more closely with business partners, the diversity of data sources today is expanding rather than contracting. This trend has been offset, to some degree, by increased reuse through the adoption of service-oriented architectures and components, but the challenge of fragmented data continues (see "Data's Role Is Not as a 'Black Box'").

Organizational trends in IT also lead to increased data fragmentation. Application support teams, data warehouse teams, integration competency centers and other groups within the IS organization claim responsibility for subsets of enterprise data. This increases the need to monitor and coordinate the enterprise's data architecture on a broader scale.

**Goals of Enterprise Data Architecture (What)**

*Understand the current data "state of affairs":* A key objective of data architecture is to help the enterprise understand what data assets it owns, where they come from, where they reside, how they are used and what their levels of data quality are. Inventorying, analyzing and documenting the architecture provides a basis for identifying opportunities for improvement. It also serves as a mechanism for communicating the architecture to all data "constituents" to raise awareness and increase leverage of existing data structures.

*Reduce data redundancy and fragmentation:* Although some data redundancy and fragmentation is inevitable, it must not be allowed to proliferate. Data redundancy is expensive, requiring additional hardware, software and support resources. The benefits of packaged applications, the risks of rewriting legacy applications and the use of extracted data should be balanced against the costs. These are business decisions that must be made within the correct governance framework (see "Seven Architecture Management Best Practices").

*Eliminate unnecessary movement of data:* Although reduced data fragmentation will lessen the need to move data, without management focus, there is a tendency for each new interface requirement to be addressed separately. The result is that, as the number of applications increase, the number of interfaces tends to increase exponentially. Over time, without active management, many of these interfaces will outlive their usefulness and become unnecessary overhead.
Develop integrative views of data: Because it is difficult, if not impossible, to eliminate all data fragmentation and redundancy, there will always be a requirement for integrated views of data to support a range of business and technical needs. Some views will be realized by creating new databases that pull together data from various sources. A number of types of data stores can do this, including data warehouses, data marts and operational data stores (see "A Taxonomy of Database Implementation Styles"). Other integrative views are oriented toward "near-real-time" access to data and involve the use of integration middleware, such as message-oriented middleware and integration brokers, or even a database gateway to directly access the source database.

Each application development team will have a different interpretation of the structure and semantics of the data. The semantic differences are often significant and obvious; although sometimes they can be subtle. However an integrative view is realized, it must reconcile these differences. For some entities, data must be gathered together from several sources in the virtual enterprise, because no single data store has the complete picture. However, it is sometimes not clear what pieces of data relate to the same situation. In many enterprises, customer data is a good example, because recognizing the same customer in different databases is not a straightforward exercise. It can even require human intervention and judgment. Data architects must selectively probe this area in considerable detail.

Reduce the number of technologies deployed: A number of benefits flow from supporting fewer technologies. Deeper discounts should be available from the vendors supplying the selected products because of the increased volume of business. With fewer products to support, the range of skills required is smaller, so fewer staff members are needed. It also allows staff to focus more on the nominated products, which enables them to deepen their skills and become more effective. As part of a data architecture effort, enterprises should review and consolidate the number of DBMS products and data movement technologies that are in place.

Improve data quality: Most enterprises are severely hampered in their efforts to perform integration and fully leverage their data because of poor data quality. A focus on data quality improvement is a critical component of the enterprise data architecture. The data architecture effort must identify data quality issues and involve business resources—not just IT resources—in identifying and implementing solutions (see "Data Stewardship: Critical Component of Data Architecture").
Improve security: Like quality, fragmented sources of data make security much more difficult to enforce. Not only must security measures be established for each data source, they must be coordinated, because inconsistent security measures may enable inappropriate access or prevent appropriate access.

Deliverables (How)

Because of its vast scope, enterprise data architecture work must be prioritized to produce timely, relevant results. One way to do this is to select a single business process and focus on the key data on which this process depends (see "Business Processes: A Compass for Architecture"). This work will often extend beyond the original business process. This selection should be done in consultation with business executives.

Specific outputs of enterprise data architecture include:

Analysis of existing data stores: These must be documented, including descriptions of how the data is organized (rows/records, primary and secondary keys), the database management system or file type, the platform, business descriptions of the data, data quality ratings and data stewards, and applications reading and updating the data.

Data models: Traditionally, many IS organizations have confused enterprise data modeling with data architecture. We do not advocate the extensive data modeling efforts that have been tried in the past, because the benefits do not outweigh the cost. However, some limited and targeted modeling is valuable to:

- Guide the development of robust and flexible canonical views (see "Managing Multiplying Messages")
- Identify and resolve semantic differences between data from different sources
- Underpin databases designs for systematic application development projects

Current and target data flows: This should cover interfaces and extracts and, like data stores, should include business descriptions of the data, the transport technology and the source and destination of the data. It should also include any transformation or consolidation that is performed en route, as well as cover service-level requirements, such as availability, volume of data, response time, and security and privacy.

Architected data stores: The target environment may require some data stores of derived data to support the strategy. These include data warehouses, data marts, operational data stores and directories that point to instances of fragmented data. For
each of these data stores, the documented information should include the logical and physical data model and the database management system, as well as the availability, throughput, response time and volume requirements.

**Infrastructure/tools standards:** The same outcome often can be achieved using different kinds of technologies. For example, data can be transformed by an application written in a language such as Java, by an extraction, transformation and loading tool, or an integration broker. Guidance on selecting the most-appropriate type of technology to perform specific tasks should be part of enterprise data architecture. Data architecture must work with technology architecture to build and maintain a list of approved products.

These deliverables are needed to achieve the data architecture goals, but they must be accompanied by management processes as described in "Seven Architecture Management Best Practices." Some data architecture deliverables are very detailed. Some deliverables are produced by other groups. The data architecture function should coordinate this work and tie it together.

**Key Facts:**

- In most enterprises, data fragmentation and redundancy is inevitable, but it prevents the enterprise from responding quickly and effectively to business challenges and opportunities.

- Enterprise data architecture addresses this by introducing structure, control and standards.

- Improvements in the effectiveness of the data architecture can provide gains in business agility and reductions in cost.

**Bottom Line:** All enterprises have a data architecture — it is the data landscape of the business. Data architecture, as a discipline, is aimed at finding opportunities to simplify, standardize and improve accessibility to data. Enterprises that focus resources on planning and executing improvements to their data architecture can realize infrastructure cost savings by removing redundancy and controlling fragmentation. Most importantly, optimization of the data architecture will increase the agility of the enterprise in responding to constantly changing business requirements.