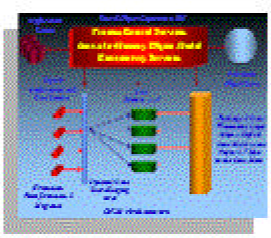
EIS



DKMS Briefs

Joseph M. Firestone

DKMS Brief No. Seven: DKMA and The Data Warehouse Bus Architecture

Introduction

The Data Warehouse Bus Architecture is composed of "a master suite of conformed dimensions" and standardized definitions of facts. [1, P. 156] Business process data marts throughout an enterprise can "plug into" this bus to receive the dimension and fact tables they need. The Bus thus supports the various processes and associated data marts that measure key aspects of the processes.

The logical union of these data marts is said to be the data warehouse. And each data mart is said to be a subset of that data warehouse. This paper describes the Data Warehouse Bus Architecture offered by Kimball, Reeves, Ross, and Thornthwaite, [1, Pp. 153-157, 266-277, and 346-347] and then contrasts it with DKM Architecture, an object-oriented alternative. [2]

The Data Warehouse Bus (DWB) Architecture

The DWB architecture pictures an enterprise composed of a set of business processes. In many businesses the processes may be components of supply chains and value chains. In other businesses they may represent "value circles." Either way, taken as a whole they represent the activities that produce the value stream of the enterprise. To describe the properties of the processes and ultimately the value stream, we develop a set of data marts, at least one for each process, the logical union of which is the data warehouse. The data marts are composed of dimension tables and fact tables. Some of these tables may be needed by more than one data mart, and more than one business process.

The tables that need to be shared must be "conformed" across data marts. The conformed dimensions are viewed by Kimball, Reeves, Ross, and Thornthwaite as constituting a bus that the business process data marts can "plug into" to receive the shared dimensions they need. The metaphor of the bus can be most clearly seen in two contexts. First, an analytical matrix [1, P. 271] whose rows are business process data marts and whose columns are dimensions is specified in the early stages of data warehouse development to guide warehouse evolution. Checkmarks in the cells of the matrix indicate where a process is "plugging into" a conformed dimension.

Second, a more visual metaphor for the DWB is provided by the system bus in a PC. [1, P. 347] Thus:

"In a way, the matrix is like the system bus in your computer. Each dimension is like a connector, or wire, on the bus. It carries a standard (i.e., conformed) 'signal' like product or customer. Each business process is an expansion card that plugs into the data connectors as appropriate. " [P. 346] It uses only those conformed dimensions that are useful for it. As new business processes are added, their associated data marts can "plug" into the existing dimensional bus.

Distributed Knowledge Management (DKM) Architecture

The DKM architecture has been outlined elsewhere. [2] [3] [4] DKM architecture is the characteristic architectural pattern of the Distributed Knowledge Management System. [5] It is an evolving O-O/Component-based architecture applicable to enterprise wide systems incorporating multiple processing styles including DSS, OLTP, and Batch processing.

Top - Down and Bottom-Up architectures may be viewed as two-tier architectures utilizing clients and local or remote databases. [2] Enterprise Data Mart Architecture (EDMA), Data Stage/Data Mart Architecture (DS/DMA), Distributed Data Warehouse/Data Mart Architecture (DDW/DMA), and the Data Warehouse Bus architecture may all be viewed as adding middleware and tuple layers to earlier architectures. The layers are added to provide the capability to manage warehouse systems integration through unified logical views, monitoring, reporting, and intentional DBA maintenance activity. But tuple-layer based management still doesn't provide automatic feedback of changes in one component of a data warehousing system to others.

DKM architecture may be viewed as adding an object layer to EDMA, DS/DMA, DDW/DMA, or DWB, to provide integration through automated change capture and management. The object layer contains process distribution services, in-memory and persistent object models, and connectivity to a variety of data store and application types. The layer requires an architectural component called an Active Knowledge Manager (AKM). [2][3][4]. Figure One provides a graphic of DKM architecture in the data warehousing context.

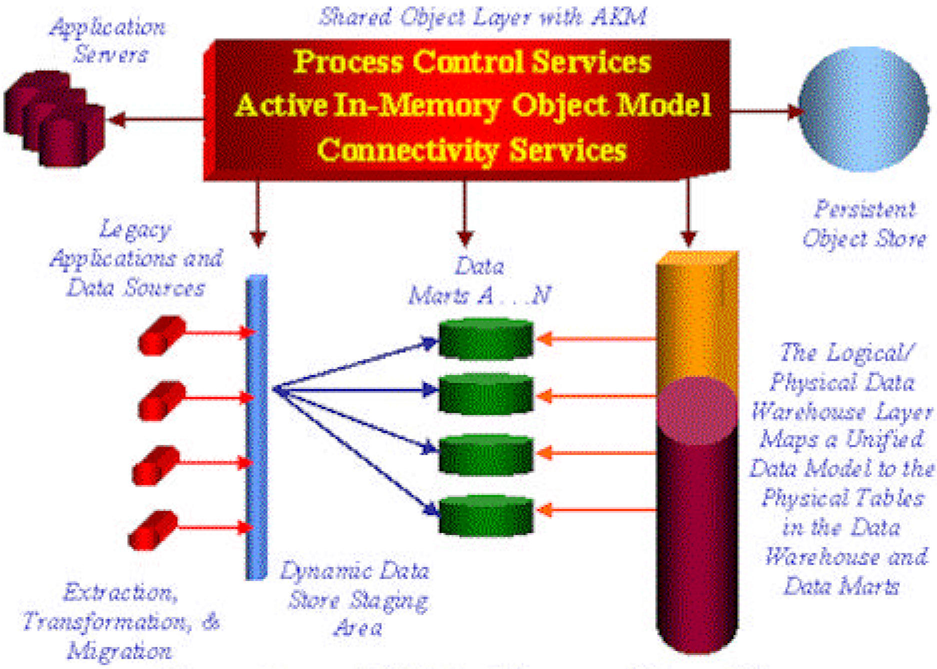

Application Servers
Shared Object Layer with AKM
Process Control Services
Active In-Memory Object Model
Connectivity Services
Persistent Object Store
Legacy Applications and Data Sources
Data Marts A . . . N
The Logical/ Physical Data Warehouse Layer Maps a Unified Data Model to the Physical Tables in the Data Warehouse and Data Marts
Extraction, Transformation, & Migration
Dynamic Data Store Staging Area

Figure One -- DKM Architecture (From [2])

The Active Knowledge Manager

An AKM provides process control and distribution services, an object model of the DKMS, and connectivity to all enterprise information, data stores, and applications. Process Control and Distribution services are the activities that manage communication among and change in the objects and components of the DKMS, as well as work flow in the system. The in-memory active object model provides the behavioral capabilities and attributes needed for the process control and distribution services. And the connectivity services provide the communications access needed to perform the process control and distribution services.

Relating the Architectures

The DKM architecture is much broader in scope than the DWB architecture. It addresses a broader set of issues, and is comparable to a data warehouse systems architecture, [See 1, P. 508] such as the "Big Shopper Data Warehouse Technical Architecture Model," rather than the more limited in scope data architecture of the DWB. But within DKM architecture there is a persistent object store. It is a repository catalog of all of the objects within the DKMS.

Among these objects are entity objects encapsulating all of the data attributes in the DKMS. Among these entity objects are dimensional objects and fact objects. [6] Moreover, process control and distribution services in the DKM architecture, guarantee both conformed dimensional and fact objects as the system evolves, and transparent object sharing across data marts when object sharing is desirable.

In short, DKM architecture embeds an Object Warehouse Bus (OWB) of conformed fact and dimensional objects. In fact, a fundamental idea of object orientation is the notion that objects support business processes and use cases. And that it is through objects that business processes are performed. [7] This idea is analogous to the emphasis of Kimball, Reeves, Ross, and Thornthwaite on data marts supporting business processes and the DWB providing a "bus' that the processes can "plug into."

So, the OWB, like the DWB provides a systematic foundation for business process data marts, and supports their construction and maintenance. Entity objects (and other types of objects as well) are permanently staged in the persistent object catalog. They may be used whenever a new data mart needs to be constructed, and they may be freely distributed across the diverse data stores and application platforms of the DKMS.

The metaphor of an electronic or PC-bus not only fits tuple layer structures such as the tables in a relational database, but also fits object layer structures such as the entity objects in the AKM's persistent data store. In addition to the data warehouse bus, there is also an object warehouse bus, for those who wish to take advantage of it.

Beyond the DWB and the OWB

The DWB architecture supports sharing dimension and fact tables, and provides consistency of data and semantics across data marts. The OWB within DKM architecture supports sharing dimensional and fact objects, consistency of data and semantics, and sharing of methods of data access across data marts. So it provides a somewhat more comprehensive shared architecture encompassing not only data aspects but access methods as well.

But DKM architecture, as I stated earlier, is much broader in scope than either DWB architecture, or the OWB, and therefore it provides a much greater breadth of architectural sharing across data marts. Specifically, in addition to sharing entity objects and access methods for those objects, DKM architecture provides for sharing control objects. And through them: (a) declarative business rules encapsulated in the objects, (b) procedural networks of business rules including computational algorithms, and analytical models, and (c) reports and query formats and results.

It also provides for tracking what is shared, so that change in any shared object or component at any point in the distributed system is automatically tracked, managed, and integrated into the system. In addition, DKM architecture provides a variety of other capabilities to all data marts in the data warehousing system.

In the area of process control and distribution services these include:

- in - memory proactive object state management and synchronization across distributed objects (including business rule management and processing, and metadata management);
- component management;
- workflow management;
- transactional multithreading;
- CORBA and/or COM messaging.

These services provide an integrative "glue," and comprehensive management capability to the data warehousing system that is not available from either the DWB or the OWB alone.

The in-memory Active Object Model and Persistent Object Store Model components of the AKM include:

- Event-driven behavior;
- DKMS-wide model with shared representation;
- Declarative business rules;
- Caching through partial instantiation; and
- A Persistent Object Store for the AKM.

The role of the object model is to provide the AKM with a shared representation of system state across all of the data marts. This state is available to all of the data marts, and when changes occur in any, the reflexive objects in the AKM specific to that data mart alert related objects in the system. While a shared state representation of entity objects is available from the OWB, and a shared representation of data content and logic is available from the DWB, these will not meet the needs of the various process control and distribution services. Only the architectural component of a shared object model including control objects providing for event-driven behavior, business rules, and partial instantiation of objects can provide the foundation needed for these.

The Connectivity Services of the AKM include:

- Language APIs: C, C++, Java, CORBA, COM
- Databases: Relational, ODBC, OODBMS, hierarchical, network, flat file, etc.
- Wrapper connectivity for application software: custom, CORBA, or COM-based.
- All legacy, relational, and object/component-based applications.

The DWB and the OWB are not broad enough in scope to provide shared connectivity to the data marts. They don't deal with the architecture of connectivity. So here is another area where DKM architecture exceeds their bounds.

In data warehousing systems based on a metadata layer, various products provide connectivity. The architecture of connectivity is specified accordingly. In the AKM though, connectivity comes integrated. It is provided by its control objects which provide access to sources of data mapped to the object model.

This broad range of connectivity services of the AKM provides it with the ability to map and connect its abstract object model to the corporate reality of diverse and chaotic data, information, and knowledge resources. The AKM, however elegant and systematic its internal organization, and however, powerful its methods for process distribution and control could not operate without the data necessary to populate its object and component-based abstractions. The extent to which the AKM's connectivity is universal, places an outer boundary on its capabilities for full data warehouse systems integration. If you can't connect to a data resource, you can't integrate it into your system.

Conclusion

The DWB provides an architecture for integrating data across data marts. It is not true though, that the architectural framework provided by the DWB alone is sufficient to provide the foundation for an incremental data warehouse. That is, the union of business process data marts is not a data warehouse. [1, Pp. 19, 200-203, 266-271] There are at least two reasons why this is not so.

First, because the union of business process data marts doesn't necessarily provide the data needed for management decision support for departments, or for departmental interactions among themselves and with the external world. So unless the term business processes includes processes internal to departments, the statement won't hold even at the level of data attributes.

Second, though, and more importantly, the DWB architecture alone, does not address process control and distribution issues, shared methods and behavior of data marts becoming integrated into a data warehouse, and shared aspects of connectivity. So if you put together a set of data marts sharing a DWB architecture in hopes of getting a data warehouse in the course of time, you'll still have to contend with conflicting processes, methods, behavior, and connectivity needs when you attempt to perform the integration. You won't have solved the islands of information problem, but only the islands of data problem.

A data warehouse is not just a data store. It is a data store capable of providing decision support. This means it must be integrated with the environment around it. And this, in turn, means that the DWB must be supplemented with a process and metadata architecture that the various data marts can hold in common. And finally, it means that this process and metadata architecture must be scalable to the enterprise level.

But these process and metadata requirements go beyond the conceptual bounds of the DWB as they do of the bounds of the OWB. To handle these requirements you need a data warehouse systems architecture, and not just a data warehouse bus architecture. Such an architecture is provided by DKM architecture and its Active Knowledge Manager component. It is the DKM architecture that provides both the data and process integration necessary for incremental, evolutionary, data warehouse construction from data marts.

References

[1] Ralph Kimball, Laura Reeves, Margy Ross, and Warren Thornthwaite, The Data Warehouse Life Cycle Toolkit (New York: John Wiley & Sons, 1998).

[2] Joseph M. Firestone,"Architectural Evolution in Data Warehousing," available at http://www.dkms.com/White_Papers.htm.

[3] Joseph M. Firestone, "DKMS Brief No. Four: Business Process Engines in Distributed Knowledge Management Systems," available at http://www.dkms.com/White_Papers.htm.

[4] Joseph M. Firestone, " DKMS Brief No. Six: Data Warehouses, Data Marts, and Data Warehousing: New Definitions and New Conceptions," available at http://www.dkms.com/White_Papers.htm.

[5] Joseph M. Firestone, "Distributed Knowledge Management Systems: The Next Wave in DSS," available at http://www.dkms.com/White_Papers.htm.

[6] Joseph M. Firestone, "Dimensional Object Modeling," available at http://www.dkms.com/White_Papers.htm.

[7] Ivar Jacobson, Maria Ericsson and Agneta Jacobson, The Object Advantage: Business Process Reengineering with Object Technology (Reading, MA: Addison-Wesley, 1995)

Biography

Joseph M. Firestone is an independent Information Technology consultant working in the areas of Decision Support (especially Data Marts and Data Mining), Business Process Reengineering and Database Marketing. He formulated and is developing the idea of Market Systems Reengineering (MSR). In addition, he is developing an integrated data mining approach incorporating a fair comparison methodology for evaluating data mining results. Finally, he is formulating the concept of Distributed Knowledge Management Systems (DKMS) as an organizing framework for the next business "killer app." You can e-mail Joe at eisai@home.com.

[Up] [Data Warehouses and Data Marts: New Definitions and New Conceptions]
[Is Data Staging Relational: A Comment]
[DKMA and The Data Warehouse Bus Architecture]
[The Corporate Information Factory or the Corporate Knowledge Factory]
[Architectural Evolution in Data Warehousing]
[Dimensional Modeling and E-R Modeling in the Data Warehouse]
[Dimensional Object Modeling] [Evaluating OLAP Alternatives]
[Data Mining and KDD: A Shifting Mosaic]
[Data Warehouses and Data Marts: A Dynamic View]
[A Systems Approach to Dimensional Modeling in Data Marts]
[Object-Oriented Data Warehousing]