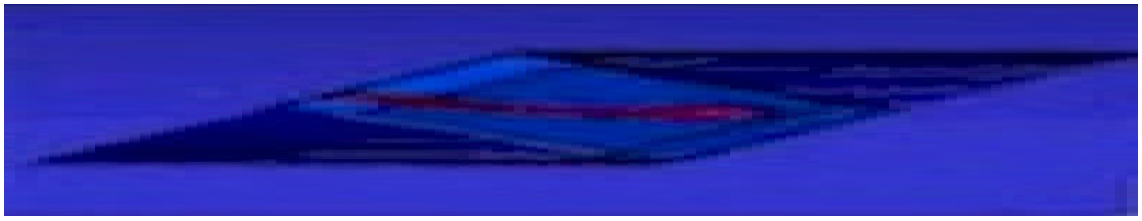

Working Paper No. Two

Knowledge Base Management Systems and The Knowledge Warehouse: A "Strawman"

By

Joseph M. Firestone, Ph.D.

Executive Information Systems, Inc.

http://www.dkms.com

eisai@home.com

Revised March 16, 2000

Prepared for First KMCI/AIIM

KM ANSI/ISO Standards Committee Meeting

January 29, 1999

This paper is a working paper, or "straw man," circulated for purposes of collaboration within the Knowledge Management Consortium International's (KMCI) Artificial Knowledge Management Systems Committee (AKMSC). It is intended that this paper be used by the Committee, along with contributions of other committee members to arrive at a collaborative Standard Recommended Practice on Artificial Knowledge Base Management Systems, a product of the Committee and the KMCI.

Introduction

The concepts of Knowledge Base Management System (KBMS) and the Knowledge Warehouse (KW) are analogues of Database Management System (DBMS) and Data Warehouse. To arrive at a standard practice on the KBMS, and a standard definition of the Knowledge Warehouse, it's reasonable to begin with "straw man" definitions of both these concepts, next develop a general concept of

what a standard practice might encompass, and then subject these products to vigorous criticism and analysis by the AKMSC. To produce this straw man is the purpose of this paper. I will proceed by considering some basic distinctions among data, information and knowledge, then discuss DBMSs, the DW, DW evolution, and Data Warehousing as a process, and then move from there to develop the analogous concepts in the knowledge and knowledge management sphere.

What are the Differences among Data, Information, and Knowledge?

To begin with, organizational data, information, and knowledge, all emerge from the social process of an organization, and are not private. In defining them, we are not trying to formulate definitions that will elucidate the nature of personal data, information, or knowledge. Instead, to use a word that used to be more popular in discourse than it is at present, we are trying to specify inter-subjective constructs and to provide metrics for them.

A datum is the value of an observable, measurable or calculable attribute. Data is more than one such attribute value. Is a datum (or is data) information? Not in itself; but information is provided by a datum, or by data, because data is always specified in some conceptual context. At a minimum, the context must include the class to which the attribute belongs, the object that is a member of that class, some ideas about object operations or behavior, and relationships to other objects and classes.

Data alone and in the abstract therefore, does not provide information. Rather, information, in general terms, is data plus conceptual commitments and interpretations. Information is data extracted, filtered or formatted in some way (but keep in mind that data is always extracted filtered, or formatted in some way).

Knowledge is a subset of information. But it is a subset that has been extracted, filtered, or formatted in a very special way. More specifically, the information we call knowledge is information that has been subjected to, and passed tests of validation. Common sense knowledge is information that has been validated by common sense experience. Scientific knowledge is information (hypotheses and theories) validated by the rules and tests applied to it by some scientific community.

More formally, the hierarchical network of the organization’s validated rules is the knowledge base of the organization or enterprise. [1] Each rule in the network relates antecedent attribute values to consequent attribute values, concepts, or rule sequences. The attributes involved belong to a number of concepts that represent the components of the model. Declarative Rule networks are those whose rules fire in parallel to determine an outcome. Procedural Rule networks are those whose rules fire in sequence. The knowledge base is composed of both declarative and procedural rule networks.

The organization's knowledge base enables it to explain, anticipate, and predict

events and interaction patterns in the organization and in its environment. The knowledge base rule network of the organization contains: its set of remembered data; its validated propositions and models (along with metadata related to their testing); its refuted propositions and models (along with metadata related to their refutation); its metamodels; and (if the system produces such an artifact) the software it uses for manipulating these.

Organizational level knowledge, in terms of this framework, is information validated by the rules and tests of the organization seeking knowledge. The quality of its knowledge then, will be largely dependent on the tendency of its validation rules and tests to produce knowledge that improves organizational performance; or in Inmon's [2, Pp. 5-11] terms: business operations, business intelligence, and business management (the organization's version of objective knowledge). [3]

From the viewpoint of the definition given of organizational knowledge, what is an organization doing when it validates information to produce knowledge? It seems reasonable to propose that the validation process is an essential aspect of the broader organizational learning process, and that validation is a form of learning. So, though knowledge is a product and not a process derived from learning, knowledge validation (validation of information to admit it into the knowledge base) is certainly closely tied to learning, and depending on the definition of organizational learning, may be viewed as derived from it.

DBMS and Related Definitions

In moving from data to DBMSs, we move from a generalized definition of data to one defined in the context of computer systems. In this context, we define a data item as "the smallest unit of named data," consisting "of any number of bits or bytes." Sometimes a data item is "referred to as a field or data element." [4, P. 12] A record is an ordered collection of named data items. Noting these definitions of data item and record, here are some common definitions of database and DBMS.

According to O'Neill, a database is: "The collection of records kept for a common purpose . . . " [5, P. 1] And a DBMS "is a program product for keeping computerized records about an enterprise." [5, P. 1]

According to C. J. Date:

"a database is: a repository for stored data. In general it is both integrated and shared. By 'integrated' we mean that the database may be thought of as a unification of several otherwise distinct data files, with any redundancy among those files partially or wholly eliminated. . . By 'shared' we mean that individual pieces of data in the database may be shared among several different users." [6, P. 4]

According to Rumbaugh, Blaha, Premerlani, Eddy, and Lorensen: "A Data Base Management System (DBMS) is a computer program for managing a permanent,

self-descriptive repository of data." [7, P. 366]

Combining various aspects of these definitions, I'll define a database as a self-descriptive, permanent, repository storing a collection of records kept for a common purpose. And a DBMS as a computer program for managing this repository. A specific DBMS programming application, is produced by using a DBMS-template to create, maintain, and enhance it. Sometimes the template software (such as Oracle, DB2, Sybase, etc.) is called a Database Management System in common usage. But we should not lose sight of the fact that the program that manages a database in any specific situation is the concrete product of using a particular template or tool for producing an actual database management application.

The Data Warehouse and Data Warehousing: Definitions and Evolution

In the beginning, there were only "islands of information: " operational data stores, legacy systems needing enterprise-wide integration, and mission-specific Decision Support Systems. Then "along came Bill" (Inmon) and his concept of the Data Warehouse (DW) (seen as the solution to the problems of information integration and redundancy) -- the embodiment of enterprise-wide DSS for the ‘90s.

Inmon defined the DW as "a

- subject-oriented
- integrated,
- time-variant
- non-volatile
- collection of data in support of management’s decision making process." [8, P. 1]

This is the classic definition of the Data Warehouse. According to it, the DW is a type of database managed by a DBMS. Indeed, in its present form the DW is a database that uses a relational DBMS. Inmon's definition is now undergoing change as the DW field evolves. Figure One depicts where DW began.

Data Marts and Data Mining were not part of the vision of Figure One. At the beginning, there was only the DW. But the vision was too sweeping. DW’s were too costly, often impolitic, took too long to implement, and their architecture turned out to be too simple to support growing customer requirements. So, evolution in data warehousing systems began with the introduction of:

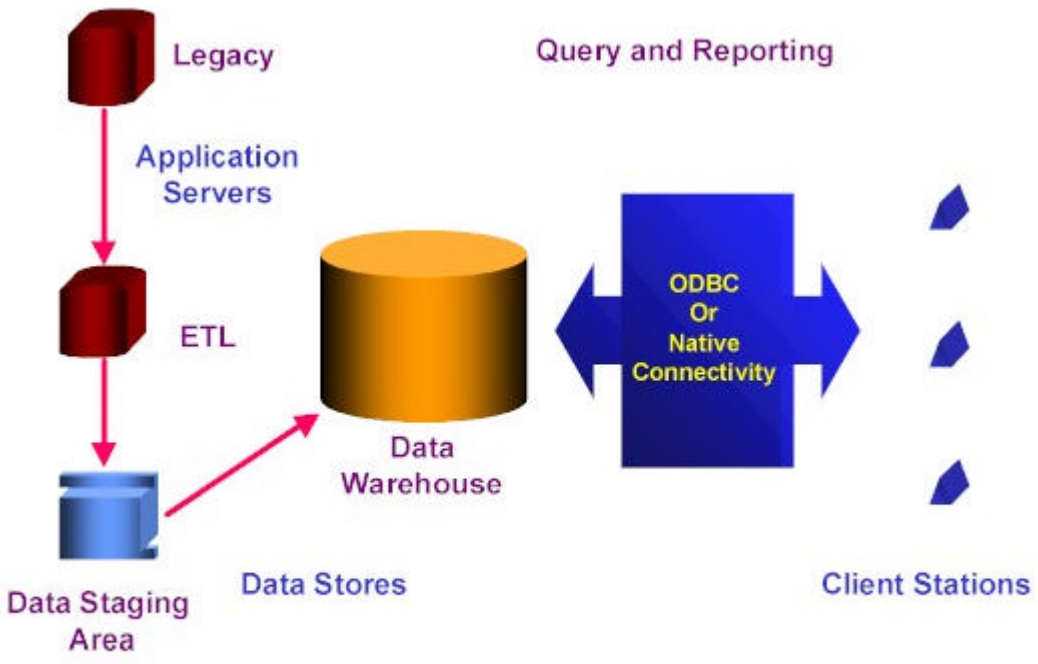

Figure One -- Where Data Warehousing Began

- Data Marts
- Dynamic Data Staging areas
- Operational Data Stores
- Web and OLAP Clients,

in response to specific customer requirements.

A variety of application servers were or are also being added to the ETML, Legacy, and Database Servers in DW systems in order to fill a variety of other user needs. Currently, intelligent agent technology is being integrated into DW systems, though we do not yet see Agency Application Servers and a generalized use of agents. The most powerful current trend is probably introduction of "second generation" metadata exchange architectures based on a commitment to DCOM or CORBA, and Object Technology for improving metadata managers and integration of ETML, other application servers and DSS data stores. Figure Two depicts where DW is now.

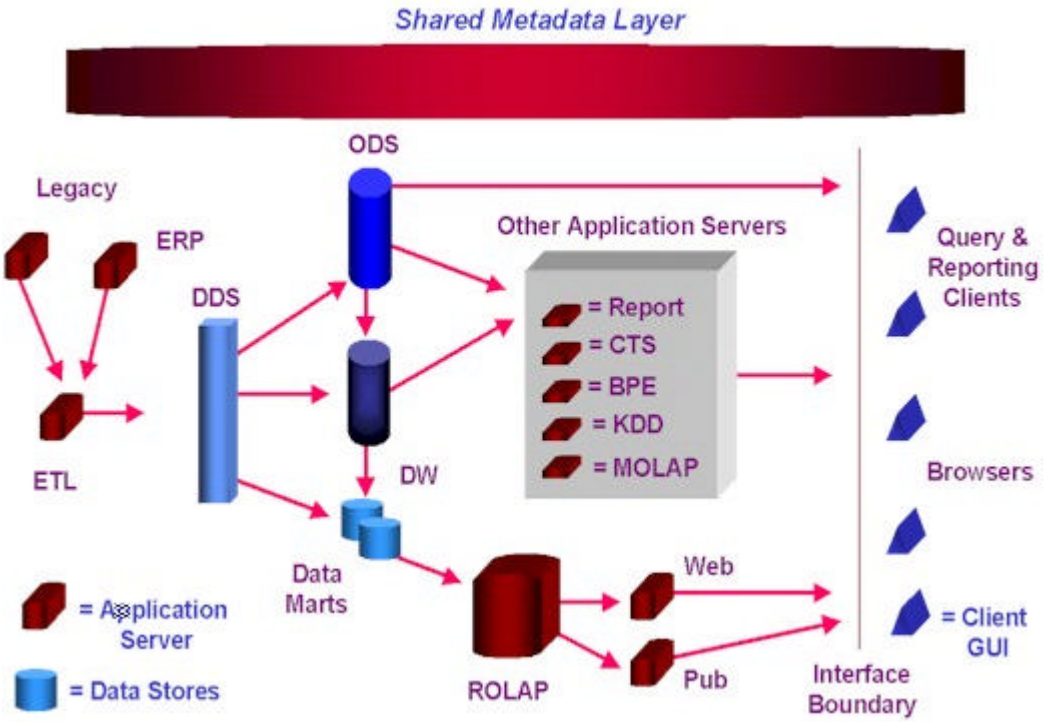

Figure Two -- Data Warehousing Now

Legend for Figures Two and Three

Web = Web Information Server

Pub = Publication & Delivery Server

KDD = Knowledge Discovery in Databases/Data Mining Servers

ETML = Extraction, Transformation, Migration and Loading

DDS = Dynamic Data Staging Area

DW = Data Warehouse

ODS = Operational Data Store

ERP = Enterprise Resource Planning

Query = Query and Reporting Server

CTS = Component Transaction Server

BPE = Business Process Engine

ROLAP = Relational Online Analytical Processing

Note the great increase in functionality and complexity in the above system, and the correspondingly greater need for integrative mechanisms. In particular, the greater and increasing role of application servers in general, and Business Process Engines (BPE) in particular, is manifest in data warehousing. Business Process Engines are application servers that maintain state in memory rather than in persistent storage. [9, P. 1] ROLAP and KDD servers are also BPEs. As reflected in Figure Two, metadata is now heavily emphasized as an integrative mechanism.

With these changes have come other definitions of the Data Warehouse and evolving conceptions of Data Warehousing These have been offered with no real attempt to confront other, different definitions or conceptions, or to explore the reasons for disagreements in definitions, and the conceptual commitments or gaps that these definitions imply. Let's look at some newer definitions, and then discuss developments in conceptions of Data Warehousing.

Newer DW Definitions

Inmon's classic definition of the DW, taken alone, does not distinguish a data warehouse from a data mart, or enterprise wide data warehouses from process-oriented data warehouses. That is, it does not distinguish subject-oriented, integrated, time-variant, non-volatile data stores of differing scope.

Inmon and his collaborators define a data mart as "a subset of a data warehouse that has been customized to fit the needs of a department." [2, P. 70] They also emphasize that "a data mart is a subset of a data warehouse, containing a small amount of detailed data and a generous portion of summarized data." There is no agreement on this, as there is a strong counter position that atomic data marts are the foundation of the data warehouse. [10, Pp. 346-348]

The following types of DSS data stores all fit the characterization "subject-oriented, integrated, time-variant, non-volatile collection of data in support of management's decision making process." They represent different concepts, but together they should provide a framework for reasoning about the issue of definition.

- A galactic data warehouse is a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management's decision making process about any and all enterprise business processes and departments, and about the enterprise taken as a whole.

- A business process-oriented data warehouse is a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management's decision making process about any and all business processes and their interactions with one another and the external world.

- A department-oriented data warehouse is a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management's decision making process about any and all departments, and their interactions with one another and with the external world.

- A business process data mart is a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management's decision making process focused on a single business process.

- A departmental data mart is a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management's decision making process focused on a single department.

This framework provides three types of data warehouses and two types of data marts. I think businesses are mostly interested in business process data warehouses and data marts. And that the initial interest in Galactic Data Warehouses has faded, while a focus on departmental data warehouses and data marts is less desirable because it is not consistent with the widely endorsed business process orientation toward increasing productivity. Other recent definitions of data warehouse are focusing on the idea that a simple part-whole relationship exists between these categories and that the union of data marts is the data warehouse.

The union of departmental data marts, however, is not a data warehouse, because this union doesn't necessarily provide management decision support for cross-departmental business processes, or for departmental interactions among themselves and with the external world. Still a departmental data mart is a subset of a galactic data warehouse or a department-oriented data warehouse as defined above.

The union of business process data marts is also not a data warehouse, as Ralph Kimball and his collaborators suggest, [10, Pp. 19, 200-203, 266-271] because this union doesn't necessarily provide management decision support for departments, or for departmental interactions among themselves and with the external world. Still a business process data mart is a subset of a galactic data warehouse or a business process-oriented data warehouse as defined above.

The above definitions of the data warehouse don't preclude the possibility that the data warehouse could be distributed. While insisting that the data warehouse is a unified, integrated logical entity, at the physical level the possibility is there that the data warehouse could be distributed.

Changes in Data Warehousing

Data Warehousing used to focus on gathering data from legacy sources of various kinds, putting it through the ETML process, loading it into the data warehouse, and providing reporting tools and report templates to access it conveniently. Given the

changes in DW system complexity, Data Warehousing is now, increasingly, a problem of integrating a variety of distributed warehouse data stores with various specialized application servers and front end access devices that need warehouse data. The Data Warehousing System, which began as a low volatility system, is now a system that may integrate DSS, batch and OLTP processing, and that therefore may incorporate considerable volatility.

The current state of Data Warehousing raises the following issues. How can increasingly complex data warehousing systems:

- achieve dynamic integration?
- comprehensively integrate and support knowledge production?
- store knowledge for high capability decision support?
- efficiently deliver tactical decision support using volatile data stores?
- integrate ERP systems?
- integrate increasingly varied business process engines?

To successfully resolve these issues, data warehousing systems need an integrative component with the capabilities of the Artificial Knowledge Manager AKM) [11], so that future Data Warehousing Systems will look like Figure Three.

Figure Three -- The Future of Data Warehousing

Artificial Knowledge Bases (AKBs), Knowledge Base Management Systems (KBMS), and Knowledge Warehouses (KW)

With this account of Data Warehousing as background let's discuss AKBs, KBMSs, and KWs. Previously we stated that the knowledge base of an organization contains: its set of remembered data; its validated propositions and models (along with metadata related to their testing); its refuted propositions and models (along with metadata related to their refutation); its metamodels; and (if the system produces such an artifact) the software it uses for manipulating these. The organization's knowledge base is an abstract phenomenon. And it is one that emerges from the interaction of the various agents comprising the organization. Measures of an organization's knowledge base may be found in its cultural artifacts [12], including its linguistic products, its electronic artifacts, and its artistic expressions, if any.

One type of cultural artifact of an organization is its Artificial Knowledge Base (AKB). An AKB is the portion of an organization's knowledge base expressed in the persistent storage and non-persistent memory of its computers. The AKB, like a database, is self-descriptive, is ultimately composed of bits and bytes, is permanent in the sense that it is an on-going system, is located both in specific in-memory locations and in specific persistent storage location, and is kept to fulfill an organization's purposes. Unlike a database which stores records, however, an AKB stores a network of objects and components, and these encapsulate data and methods (validated and unvalidated procedural or declarative rules that use validated and unvalidated data). So the AKB stores data and information as well as knowledge.

A Knowledge Base Management System (KBMS) is a computer application for managing (creating, enhancing, and maintaining) the AKB, just as a DBMS is a computer application for managing a database. But what does such a computer program do? To answer this question, return to Figure Three.

Figure Three is not simply a Data Warehousing System. It is an Enterprise Artificial Knowledge Management System (AKMS) as defined in Working Paper No. One. [11] It is also a KBMS, because it can produce and manage (through the AKM, its database management, application server, Knowledge Discovery in Databases/Data Mining application server, and communications and connectivity software) not only data and information, but also the network of objects and components constituting an AKB. Thus, the KDD/Data Mining Server is a component for supporting production of validated information (or knowledge). And the persistent data stores in Figure Three are not simply data stores, but taken together, including their OODBMS component, they are knowledge stores. They can store objects, and methods, and rules, and validation information, as well as data. And that makes Figure Three a Knowledge Warehousing System, and not just a Data Warehousing System.

In a nutshell, the changes summarized above, indicate that data warehousing systems are about to evolve into AKMSs, or equivalently, Knowledge Base Management Systems, or Knowledge Warehousing Systems, and that

convergence between data warehousing, DSS, and KM is about to occur. There is no separate Knowledge Base Management System. The KBMS is both the AKMS and the Knowledge Warehousing System. Take your pick on the name.

What of the Knowledge Warehouse? Like the DW, it may be viewed as subject-oriented, integrated, time-variant, and supportive of management's decision making processes. But unlike the DW, it is a combination of volatile and non-volatile objects and components, and, of course, it stores not only data, but also information and knowledge.

The KW is not co-extensive with the AKMS. It is also not a physical subsystem of the AKMS, as the data warehouse is of the DSS it supports, to which one can easily point. The KW is physically resident both in-memory and in persistent distributed data stores. Abstractly, however, the KW is the AKB itself. There is no distinction between the AKB and the KW, as there is between an enterprise wide federated database, and its data warehouse component.

The AKMS is an On-line Complex Processing (OLCP) System. It is not merely a DSS system, like today's data warehousing system. Nor is it an OLTP system, like today's ERP systems. The AKMS, given present technology, is a distributed processing system, or as I have called it elsewhere a Distributed Knowledge Management System (DKMS). [13] Since the KBMS is the AKMS, it follows that the standard the AKMSC is developing for the AKMS, is also the KBMS standard, and any software tools developed on the basis of the standard will be KBMS tools as well as AKMS tools.

On the subject of tools, there are no analogues to DBMS templates available for developing AKMSs. Such tools would need to provide templates for creating persistent data stores, for in-memory object models, for broad connectivity of the AKM to applications, databases, client modules, and communications buses. Current tools come close to having that broad range of capability, and it is possible to constitute a "best-of-breed" suite for constructing AKMSs. But I don't know of a single vendor that provides a tool suite in all of these areas.

A Standard Recommended Practice for the KBMS

If the KBMS and the AKMS are one and the same, and the KW and the AKB are also equivalent, it follows that the standard recommended practice for the KBMS is the same as the standard recommended practice for the AKMS. To develop such a recommendation, we first need to define the AKMS standard in much greater detail. To do this we need to implement the AKMSC "straw man" program outlined in Working Paper No. One. Here again is the list of tasks in the program.

1. Specify AKMS Use Case Model and Relate to NKMS Processes and Activities

2. Specify the Artificial Knowledge Manager (AKM) Logical Component

3. Specify Types of Client Application Components.
4. Specify Types of Application Servers
5. Specify Communication Buses including Object Request Brokers (ORBs)
6. Specify Types of Data Stores
7. Specify AKMS Architectural Model
8. Specify AKMS Model
9. Specify Artificial Knowledge Manager Standard
10. Specify Artificial Knowledge Base/Knowledge Warehouse Standard

Once the AKMS standard is developed, we can proceed to develop the standard recommended practice for implementing an AKMS. In the mean time, fields likely to contribute to the standard can be studied. The two main ones are Object-Oriented Software Engineering (OOSE) and Data Warehousing. Both fields are in ferment right now, and practitioners and vendors alike are offering methodologies for their Communities of Practice (CoP). In OOSE, methodologies utilizing the Unified Modeling Language (UML), aimed at rapid application development of distributed object applications are now beginning to appear. In data warehousing, the simplistic methodologies of the early days of two tier data warehousing are giving way to incremental, iterative methodologies for developing distributed data warehouses over time. The standard recommended practice for the AKMS may perhaps be developed as a synthesis of these two developing CoPs.

References

[1] Compare Edward Swanstrom, "What is Knowledge Management?" Discussion Rough Draft of a Chapter undergoing editing by John Wiley & sons, available at http://www.km.org/introkm.html.

[2] See W. H. Inmon, Claudia Imhoff, and Ryan Sousa, Corporate Information Factory (New York, NY: John Wiley & Sons, 1998).

[3] I read Gene Bellinger's views on data, information, knowledge, and wisdom at http://www.radix.net/~crbnblu/musings/kmgmt/kmgmt.htm, before writing my own differing account of the previous concepts. His views are certainly worth keeping in mind when considering mine

[4] James Martin, Computer Data-base Organization (Englewood, Cliffs, NJ: Prentice-Hall, 1977)

[5] Patrick O'Neill, Database: Principles, Programming, Performance (San

Francisco, CA: Morgan Kaufmann, 1994).

[6] C. J. Date, An Introduction to Database Systems, Vol. I (Reading, Mass.: Addison-Wesley, 1981).

[7] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, and William Lorensen, Object-Oriented Modeling and Design (Englewood Cliffs, N.J.: Prenctice-Hall, 1991).

[8] W. H. Inmon, "What is a Data Warehouse?" Prism Tech Topic, Vol. 1, No. 1, 1995

[9] John Rymer, "Business Process Engines, A New Category of Server Software, Will Burst the Barriers in Distributed Application Performance Engines," Emeryville, CA, Upstream Consulting White Paper, April 7, 1998, at http://www.persistence.com/products/wp_rymer.htm.

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

[11] Joseph M. Firestone, "The Artificial Knowledge Manager Standard: A "Strawman"" Working Paper No. One, Gaithersburg, MD: Knowledge Management Consortium, January 25, 1999.

[12] See Joseph M. Firestone, " Distributed Knowledge Management Systems and Enterprise Knowledge Management Modeling," at http://www.dkms.com/White_Papers.htm.

[13] The AKMS concept developed here is largely based on the DKMS concept I introduced in "Object-Oriented Data Warehousing," available at http://www.dkms.com/White_Papers.htm. Other papers developing various aspects of the DKMS are: Joseph M. Firestone, "Distributed Knowledge Management Systems: The Next Wave in DSS," Joseph M. Firestone,"Architectural Evolution in Data Warehousing," Joseph M. Firestone, "Knowledge Management Metrics Development: A Technical Approach," Joseph M. Firestone, "DKMS Brief No. Four: Business Process Engines in Distributed Knowledge Management Systems," all are available at http://www.dkms.com/White_Papers.htm, as are additional papers about the DKMS.

Biography

Joseph M. Firestone, Ph.D. is an Information Technology consultant working in the areas of Decision Support (especially Enterprise Knowledge Portals, Data Warehouses/Data Marts, and Data Mining), and Knowledge Management. He is consulting in the areas of developing Enterprise Information/Knowledge Portal Products, and is the author of "Approaching Enterprise Information Portals," a comprehensive,

full-length industry report on this rapidly emerging field. In addition, he formulated and is promoting the concept of Distributed Knowledge Management Systems (DKMS) as an organizing framework for software applications supporting Natural Knowledge Management Systems. Dr. Firestone is Chief Scientist of Executive Information Systems, Inc. (EIS), and one of the founding members of the Knowledge Management Consortium, International. A sampling of his writings may be found at the EIS web site at http://www.dkms.com, a site Dr. Firestone developed. The dkms.com web site is one of the more popular sites in data warehousing and knowledge management, and has now attained a run rate of more than 70,000 visits per year.