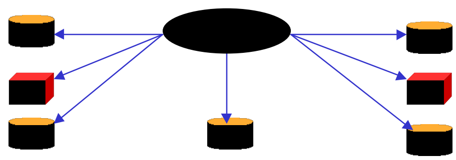
Executive Information Systems, Inc.

Architectural Evolution in Data Warehousing: The Coming of Distributed Knowledge Management Architecture (DKMA)

By

Joseph M. Firestone, Ph.D.
eisai@moon.jic.com

September 9, 1998



EIS Web Site: http://www.dkms.com

Dynamic Integration and Data Warehousing

- The Dynamic Integration Problem is the problem of proactively and automatically monitoring and managing evolutionary change in data warehousing systems without imposing a traditional and constraining “Top-Down” architecture.
- It is the problem of providing managers of both data warehouses and data marts the power to innovate, while still maintaining the integration and consistency of the system.

Dynamic Integration and Architectural Evolution in Data Warehousing

- Data Warehousing is now a complex systems integration problem. A full-blown Data Warehousing System may encompass
  - the following database servers:
    - The data warehouse
    - various data marts (department, function, or application-specific DSSs, using Relational, Multi-dimensional (MOLAP), or Column-based Servers)
    - One or more Operational Data Stores (ODSs)
    - One or more Data Staging Areas

Dynamic Integration and Architectural Evolution in Data Warehousing (Two)

- the following application servers
  - Web Servers
  - ETML Servers
  - Data Mining servers
  - Stateless Transaction Servers (e.g., MTS, Jaguar CTS, etc.
  - Business Process engines (e.g. Persistence Power-Tier, DAMAN InfoManager)
  - Document Servers
  - ROLAP Support Servers (e.g., MicroStrategy, Information Advantage)
  - Report Servers
- and various front-end OLAP and reporting tools

Dynamic Integration and Architectural Evolution in Data Warehousing (Three)

- The Dynamic Integration problem in the context of this complexity of components and interactions is three-fold:
  - First, we need an integrated view of all server-based assets;
  - Second, we need to manage flows of data, information, and knowledge throughout this system to maintain the common view in the face of change in form and content, and to distribute the system's data, information, and knowledge bases as required, and

Dynamic Integration and Architectural Evolution in Data Warehousing (Four)

- Third, we need such management to occur automatically and without centralizing the system so that the authority and responsibility for adding new data and information to the system is distributed.
- Automated dynamic integration is a capability not now provided by data warehousing vendors. It is increasing recognition of the need for this capability that drives architectural evolution in the DW system.

Architectures for Managing DSS Integration

- Here are six architectures for managing and viewing the problem of integration.
- Top-Down Architecture
  (Inmon, Prism Solutions, Carleton Corporation, ETI)
- Bottom-Up Architecture
  (Broadbase Information Systems, Sagent, Ardent DataStage)
- Enterprise Data Mart Architecture (EDMA)
  (Informatica, Carleton)
- Data Stage/Data Mart Architecture (DS/DMA)
  (Informatica, Carleton, Sagent)

Architectures for Managing DSS Integration (Two)

- Distributed Data Warehouse/Data Mart Architecture (DDW/DMA) *(Sybase, Platinum (HP) Intelligent Warehouse)*
- Distributed Knowledge Management Architecture (DKMA) *(The sole vendor targeting this architecture is DAMAN Consulting)*
- *There are variations of each architecture incorporating an ODS*

Top-Down Architecture

- The first Data Warehousing Systems architecture
- Begins with Extraction, Transformation, Migration, and Loading (ETML) process
- Establishes the data warehouse first, along with centralized metadata repository
- Data Marts are constituted from extracted and summarized data warehouse data and metadata
- The Data Warehouse has atomic layer and detailed historical data

Top-Down Architecture (Two)

- The Data Warehouse uses Normalized E-R Models
- Data Marts have highly and lightly summarized data
- Integration is automatic as long as the discipline of constituting data marts as subsets of the data warehouse is maintained
- Tools exist to generate data marts from the data warehouse “by pushing a button”

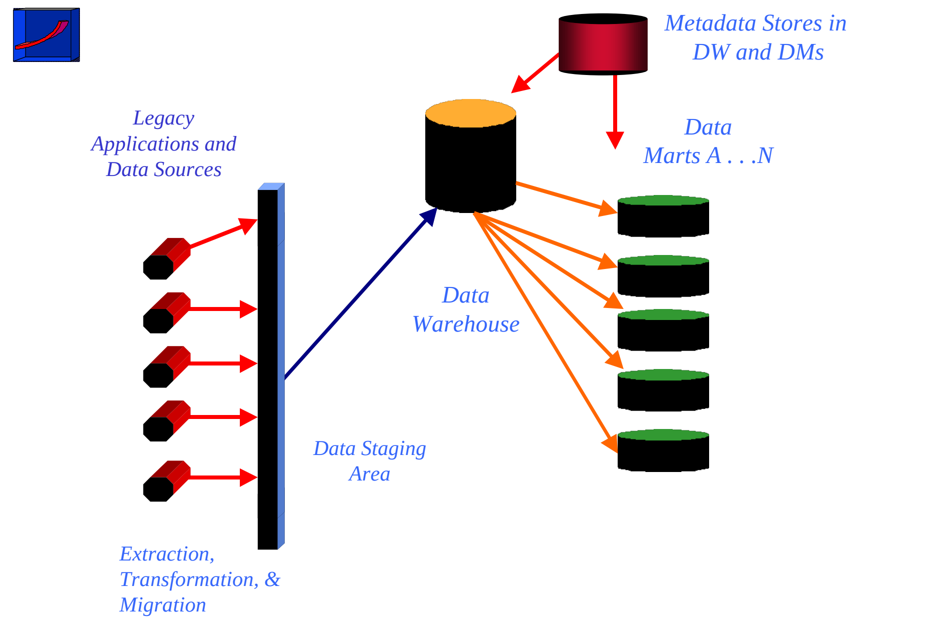


Figure One -- Top-Down Architecture

Bottom-Up Architecture

- The second Data Warehousing Systems architecture
- Became popular because the Top-down architecture took too long to implement, was often politically unacceptable, and was too expensive
- Begins with ETML for one or more Data Marts
- Requires no common data staging area
- Uses Dimensional Data models for Data Marts
- Uses Atomic, lightly summarized, and highly summarized data

Bottom-Up Architecture (Two)

- Provides no common metadata
- Constructs thedata warehouse incrementally over time from data marts
- Adopted initially by second generation tool vendors Informatica, Sagent, and Ardent
- Met expectations in building data marts, but soon was perceived as unacceptable for the long term because lacking common metadata, it is difficult to construct the data warehouse, and it also leads to new “stovepipes” or”legamarts” over time.

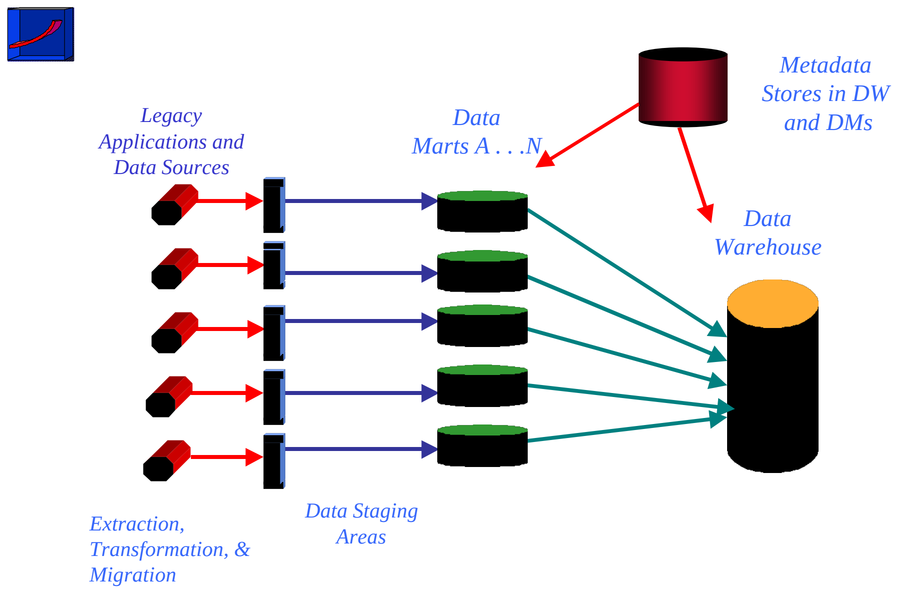


Figure Two -- Bottom-Up Architecture

Enterprise Data Mart Architecture (EDMA)

- A response of the Bottom-up supporters to "legamart" argument
- Begins with ETML for one or more data marts
- Establishes a common staging area called a Dynamic Data Store (DDS) for ET results, including a common or global metadata repository
- Constructs EDMA before beginning data marts
- EDMA includes enterprise subject areas, and common dimensions, metrics, business rules, sources, all represented in a logically common (but not necessarily physically centralized) metadata repository

Enterprise Data Mart Architecture (EDMA) (Two)

- Uses Dimensional Data Models for Data Marts
- Develops the data warehouse and data marts from the metadata repository, data marts and the DDS using an incremental approach
- Informatica's PowerCenter Tool was the first to implement this architecture. Informatica supports metadata management through monitoring and reporting mechanisms, not through an automated process. Informatica makes some use of Object Technology. Carleton is also close to this capability.

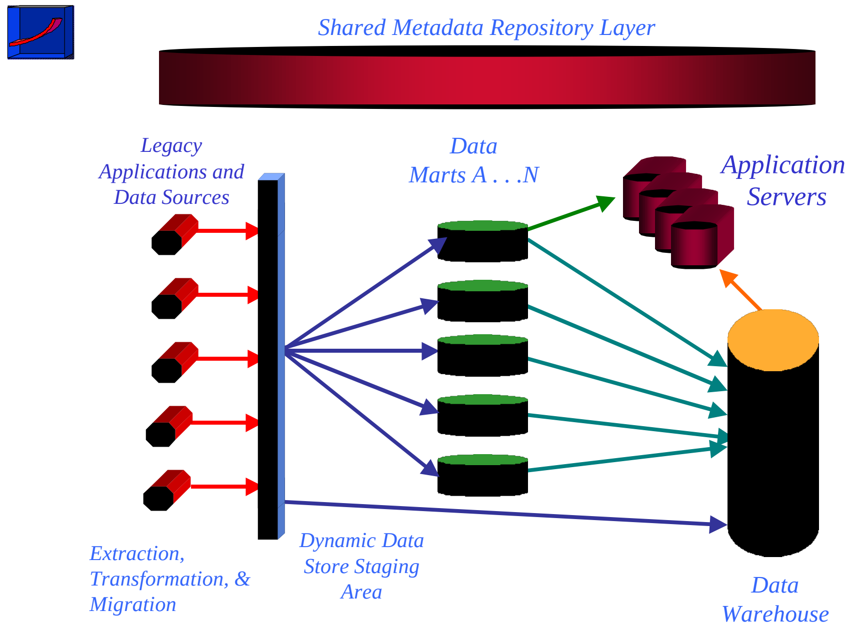


Figure Three -- EDM Architecture

Data Stage/Data Mart Architecture (DS/DMA)

- The same as EDMA with the important exception that no physical enterprise-wide data warehouse is implemented
- Instead, the data warehouse is viewed as the conjunction of the data marts in the context of an EDMA-like metadata repository
- The repository provides a common view of enterprise DSS resources, but not necessarily an enterprise-wide view, because there is no

Data Stage/Data Mart Architecture (DS/DMA) (Two)

- guarantee that the conjunction of data marts will provide access to enterprise-wide global attributes, as would a data warehouse
- Leading tool providers are again Informatica and Carleton, and also Sagent which is advocating DS/DMA in association with Ralph Kimball

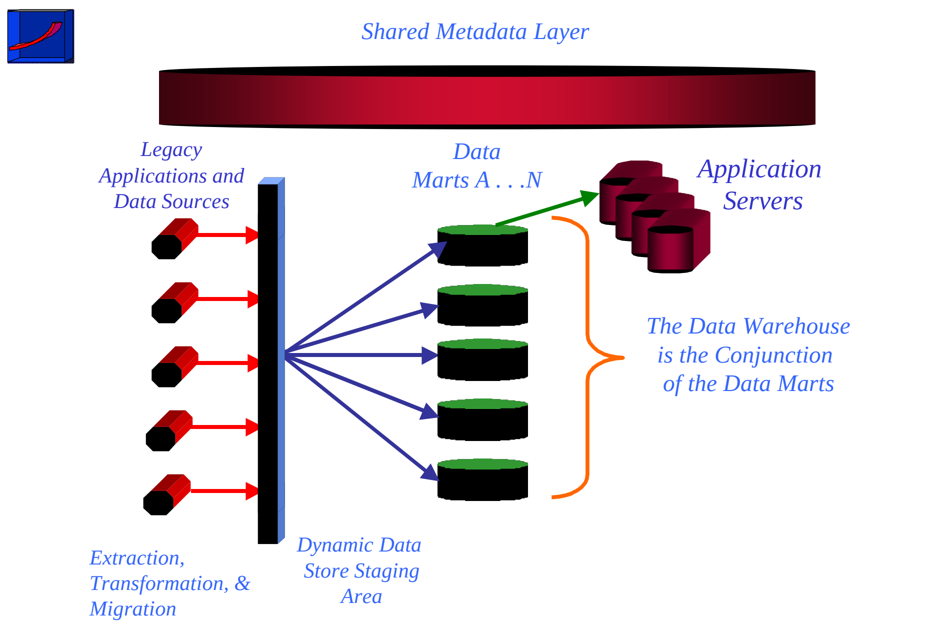


Figure Four -- DS/DM Architecture

Distributed Data Warehouse/Data Mart Architecture (DDW/DMA)

- Again similar to EDMA. Also:
- Provides common view of metadata across the enterprise
- Provides a logical database layer mapping a unified logical data model to physical tables in the various data marts and the data warehouse
- Provides transparent querying of a unified logical database across data marts along with caching and joining services.

Distributed Data Warehouse/Data Mart Architecture (DDW/DMA) (Two)

- Leading tool providers are Informatica, Carleton, Sybase Adaptive Server, and HP (now Platinum) Intelligent Warehouse
- These tools (except IW) are all offered as part of Sybase’s Warehouse Studio
- This is the most adaptable of the architectures discussed to this point, but it still reflects the limitations of the relational viewpoint when it comes to handling objects and processes, and it still doesn’t support distributed and automated change capture and management

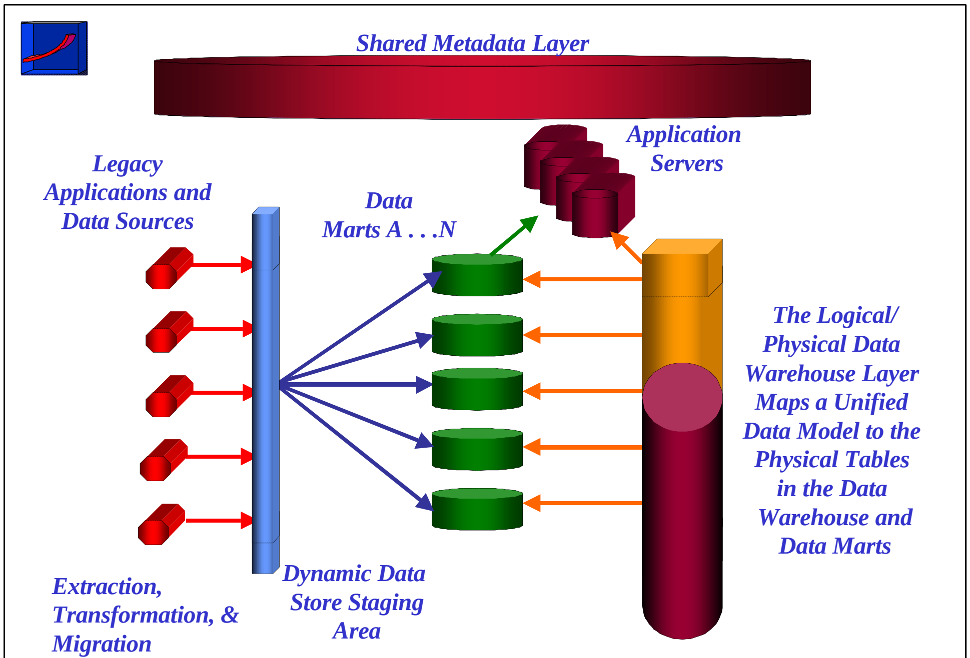


Figure Five -- DDW/DM Architecture

Distributed Knowledge Management Architecture (DKMA)

- An evolving O-O/Component-based architecture
- Top - Down and Bottom-Up architectures may be viewed as two-tier architectures utilizing clients and local or remote databases
- EDMA, DS/DMA, and DDW/DMA may be viewed as adding Middleware and Tuple layers to earlier architectures to provide the capability to manage warehouse systems integration through unified logical views, monitoring, reporting, and intentional DBA maintenance activity. But this

Distributed Knowledge Management Architecture (DKMA) (Two)

- form of management still doesn't provide automatic feedback of changes in one component to others
- DKMA may be viewed as adding an object layer to EDMA or to DDW/DMA to provide integration through automated change capture and management

Distributed Knowledge Management Architecture (DKMA) (Three)

- The object layer contains process distribution services, an in-memory active, object model, and connectivity to a variety of data store and application types. The layer requires an architectural component called an Active Knowledge Manager (AKM).

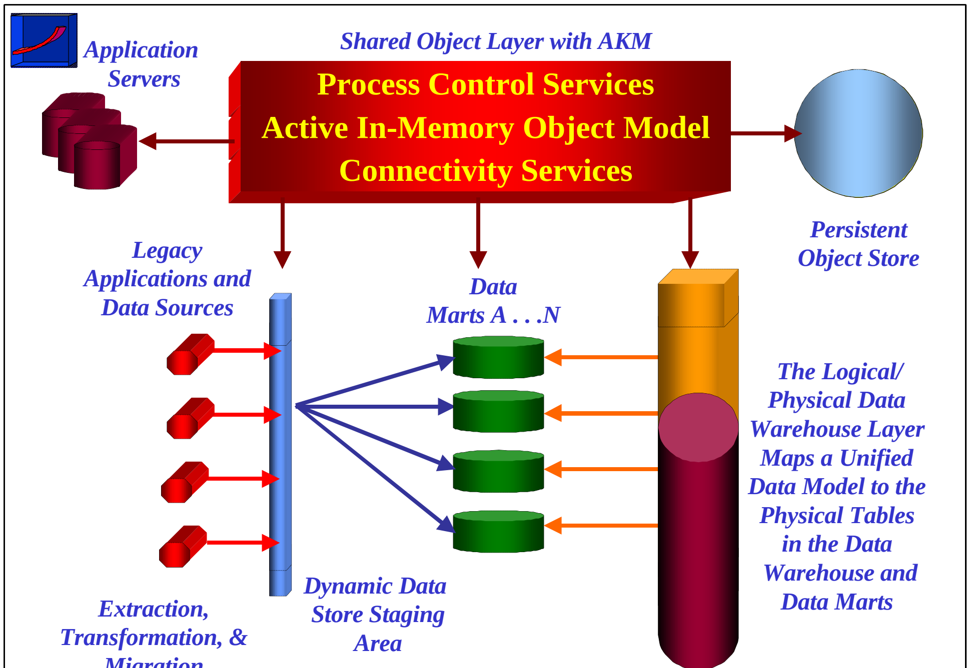


Figure Six -- DKM Architecture

DKM Architecture and the AKM

- An AKM provides Process Control Services, an Object Model of the Distributed Knowledge Management System (DKMS) (the system corresponding to the DKM architecture), and connectivity to all enterprise information, data stores, and applications
  - Process Control Services:
    - In memory proactive object state management and synchronization across distributed objects
    - Component management
    - Workflow management
    - Transactional multithreading

DKM Architecture and the AKM (Two)

- In-memory Active Object Model/Persistent Object Store is characterized by:
  - Event-driven behavior
  - DKMS-wide model with shared representation
  - Declarative business rules
  - Caching along with partial instantiation of objects
  - A Persistent Object Store for the AKM
  - Reflexive Objects
- Connectivity Services should have:
  - Language APIs: C, C++, Java, CORBA, COM
  - Databases: Relational, ODBC, OODBMS, hierarchical, network, flat file, etc.

DKM Architecture and the AKM (Three)

- Wrapper connectivity for application software: custom, CORBA, or COM-based.
- Applications include all the categories mentioned in the earlier discussion of the Dynamic Integration problem, whether these are mainframe, server, or desktop - based.

- The DKM Architecture and the AKM provide the solution to the Dynamic Integration Problem, because only the DKMA among the preceding architectures supports distributed proactive monitoring and management of change in the web of data warehouse, data mart, web information

DKM Architecture and the AKM (Four)

- servers, component transaction servers, data mining servers, ETML servers, other application servers, and front-end applications comprising today’s Enterprise DSS/Data Warehousing System.

ODS Variations

- Each of the architectures covered may vary with the addition of an Operational Data Store (ODS)
- According to Inmon: “An ODS is a collection of data containing detailed data for the purpose of satisfying the collective, integrated operational needs of the corporation . . . The ODS is:
  - subject-oriented,
  - integrated,
  - volatile,
  - current-valued,
  - detailed.”

ODS Variations (Two)

- The ODS is like a data warehouse in its first two characteristics, but it is like an OLTP system in its last three characteristics. Its purpose is to support operational, tactical decisions
- The workload of an ODS involves four kinds of processing: loading data, updating, access processing, and DSS-style analysis across many records.
- The four types of ODS processing are the source of difficulties in optimizing ODS processing. It is difficult to optimize performance over all four types.

ODS Variations (Three)

- Look at the above architectures in relation to the ODS. It is clear that an architecture that will support both DSS and OLTP-style processing is needed in order to optimally integrate the ODS into the broader data warehousing architecture. In particular,
  - process control services will be very important for the OLTP-style of processing we find in the ODS.
  - Also, distribution of ODS objects across multiple servers will help ODS performance.

ODS Variations (Four)

- Finally, in-memory processing in distributed AKMs can do much to upgrade performance in a distributed ODS.

- **Of course, only one of the above architectures can provide these capabilities for the ODS: the DKM Architecture.**

DKM Architecture and Data Mining

- A key emerging capability in DKMS and data warehousing systems is Knowledge Discovery in Databases (KDD) or Data Mining.
- The key mechanism for KDD is the data mining server.
- Here are some difficulties with current data mining server products:
  - It's difficult to incorporate new data mining algorithms, and therefore keep pace with new developments coming out of the research world;

DKM Architecture and Data Mining (Two)

- Many products require that data must be transported to proprietary data stores before data mining can occur;
- Models produced by the data mining algorithms are not freely available to power users unless they use the data mining tool itself,
- It is difficult to incorporate validation criteria not initially incorporated in the data mining tool into the KDD process,
- There are few “open architecture” commercial data mining tools.

DKM Architecture and Data Mining (Three)

- To solve these problems a product class called An Analytical Data Mining Workbench (ADMW) should be developed.
- The ADMW needs:
  - Easy and convenient encapsulation of new algorithms into object model classes;
  - Capability to mine data from any data source in the enterprise;
  - Incorporation of analytical models into an object model repository;
  - A modifiable validation model,

DKM Architecture and Data Mining (Four)

  - Integration of legacy data mining applications with the ADMW.
- An ADMW with these capabilities would meet all of the difficulties specified above.
- The DKM Architecture can help in developing the above because:
  - New algorithms can be encapsulated in objects through the "wrapping" capabilities of the AKM;
  - Data can be brought into the AKM's in-memory object model for data mining without relocating it from its data store ("chunks," partial instantiation),

DKM Architecture and Data Mining (Five)

- Data mining can be performed by executing the analytical models in memory on data chunks and partially instantiated objects;
- Analytical Models produced by an AKM -based application would be placed in an object model repository where they can be accessed by Power Users;
- Customized validity criteria could be added by modifying the validation model in the repository, because the validation model is just another object whose attributes and methods can be modified ;

DKM Architecture and Data Mining (Six)

- Legacy data mining applications could be integrated using AKM connectivity services which would “wrap” them.

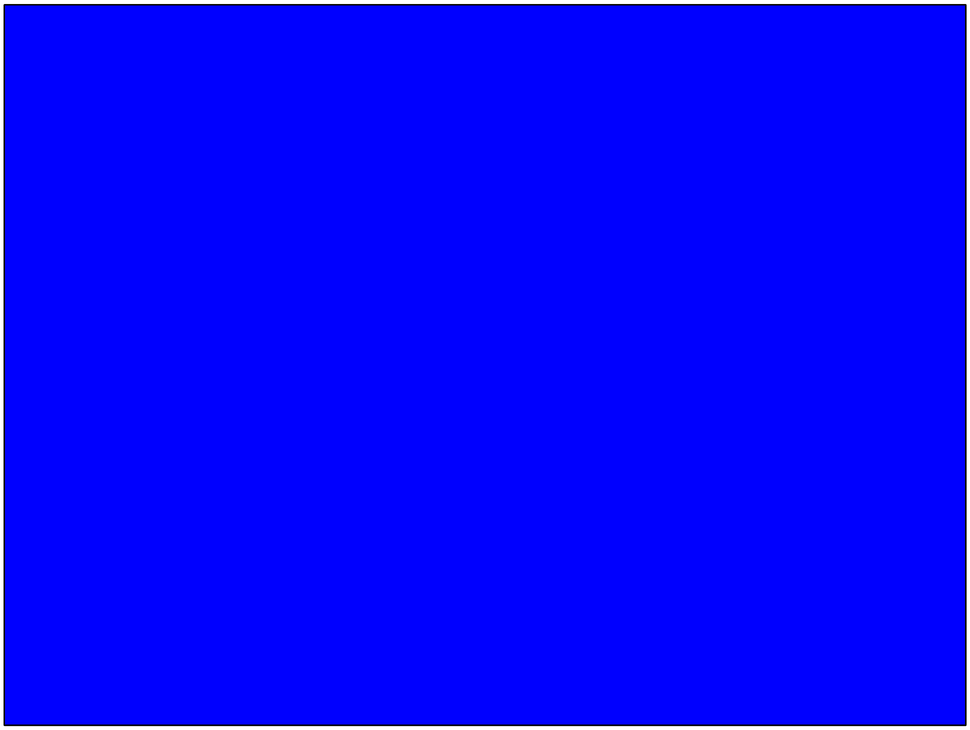
- When viewing the above, keep in mind that there is no ADMW with the above capabilities at present. Data Mining is a rapidly growing field, but the market niche represented by the ADMW is empty.
- On the other hand, there are software tools that can be used as a foundation to rapidly develop the ADMW as a facility within the AKM.

DKM Architecture and Software Tools

- To implement DKM Architecture in a DKMS you need the full range of tools now used to create data warehousing systems. In addition though, you need additional tools for the AKM component (including the ADMW facility and the ability to integrate the ODS into the DKMS). These include:
  - An object modeling RAD environment providing extensive process control services and connectivity (e.g. DAMAN's InfoManager, Template Software’s Enterprise Integration Template (EIT), Forte, a combination of Ibex's DAWN workflow product along with its Itasca Active Object

DKM Architecture and Software Tools (Two)

- Database, a combination of Rational Rose, Persistence Power-Tier; and Iona’s Orbix)
- Technology for constructing software agents to proactively monitor components of the DKMS (e.g. CA Unicenter TNG, ObjectSpace’s Voyager, Persistence Power-Tier, DAMAN’s InfoManager)
- An OODBMS to serve as a persistent object repository for the AKM component (ObjectStore, Objectivity/DB. Jasmine, Versant, Itasca)



Back-Up Slides

- Distributed Knowledge Management Systems (DKMS
- Why DKMS?
- What is the Knowledge Management System (KMS)?
- The Knowledge Base and Knowledge
- The Knowledge Management Process and Knowledge Management
- Data, Information, Knowledge, and Wisdom
- Organizational Knowledge

Distributed Knowledge Management Systems (DKMS)

- A DKMS is a system that manages the integration of distributed objects into a functioning whole producing, maintaining, and enhancing a business knowledge base.
- A business knowledge base is the set of data, validated models, metamodels, and software used for manipulating these, pertaining to the enterprise, produced either by using a DKMS, or imported from other sources upon creation of a DKMS. A DKMS, in this view, requires a knowledge base to begin operation. But it enhances its own knowledge base with the passage of time because it is a self-correcting system, subject to testing against experience.
- The DKMS must not only manage data, but all of the objects, object models, process models, use case models, object interaction models, and dynamic models, used to process data and to interpret it to produce a business knowledge base. It is because of its role in managing and processing data, objects, and models to produce, enhance, and maintain a knowledge base that the term Distributed Knowledge Management System is so appropriate.

Why DKMS?

- Other reasons for adopting the term DKMS include:
  - business knowledge production and management is what business intelligence is all about;
  - DKMS plays off DBMS, and therefore capitalizes on a familar term while favorably contrasting with it, i.e. knowledge management is clearly better than mere data management;
  - DKMS also highlights the point that data is not knowledge, but only a part of it;
  - "DKMS" is a product/results-oriented name likely to appeal to business decision makers (that is, they get valuable and valid knowledge that they can use to gain control and produce results);

What is the Knowledge Management System (KMS)?

- The Knowledge Management System (KMS) is the on-going, persistent interaction among agents within a system that produces, maintains, and enhances the system's knowledge base. This definition is meant to apply to any intelligent, adaptive system composed of interacting agents.
- An agent is a purposive, self-directed object.
- Knowledge Base will be defined in the next section.
- In saying that a system produces knowledge we are saying that it (a) gathers information and (b) compares conceptual formulations describing and evaluating its experience, with its goals, objectives, expectations or past formulations of descriptions, or evaluations.
- Further, this comparison is conducted with reference to ***validation criteria***. Through use of such criteria, intelligent systems distinguish competing descriptions and evaluations in terms of closeness to the truth, closeness to the legitimate, and closeness to the beautiful.

What is the Knowledge Management System (KMS)? (Two)

- In saying that a system maintains knowledge we are saying that it continues to evaluate its knowledge base against new information by subjecting the knowledge base to continuous testing against its validation criteria. We are also saying that to maintain its knowledge, a more complex system must ensure both the continued dissemination of its currently validated knowledge base, and continued socialization of intelligent agents in the use and content of its knowledge base.

- Finally, in saying that a system enhances its knowledge base, we are saying that it adds new propositions and new models to its knowledge base, and also simplifies and increases the explanatory and predictive power of its older propositions and models. That is, one of the functions of the KMS is to provide for the growth of knowledge.

The Knowledge Base and Knowledge

- A system's knowledge base is: the set of remembered data; validated propositions and models (along with metadata related to their testing); refuted propositions and models (along with metadata related to their refutation); metamodels; and (perhaps, if the system produces such an artifact) software used for manipulating these, pertaining to the system and produced by it.
- A knowledge management system, in this view, requires a knowledge base to begin operation. But it enhances its own knowledge base with the passage of time because it is a self-correcting system, and subjects its knowledge base to testing against experience.
- Finally, the emphasis on a system's knowledge base, rather than its knowledge, recognizes that an identification of knowledge as individual conceptions, propositions, or models is inconsistent with the reality that acceptance of a piece of information into a system's body of knowledge is dependent on the background knowledge already within the knowledge base. This background knowledge is used to filter and interpret the information being evaluated.

The Knowledge Base and Knowledge (Two)

- This definition of knowledge base contrasts with a popular definition of knowledge as "justified, true belief." The definition does agree with the necessity of justification as a necessary condition for knowledge; but it insists that justification be specific to the validation criteria used by a system to evaluate its descriptions and evaluations. The definition also agrees that knowledge is a particular kind of belief, provided that belief extends beyond cognition alone, to evaluation.

- The biggest discrepancy with the popular definition is in not requiring that justified beliefs be "true." Truth can be used as a regulating ideal by a system producing descriptive knowledge. "Right" can be used as a regulating ideal by a system producing evaluative or normative knowledge. But the system in question can never say for sure that a proposition or a model within its knowledge base is "true," or "right;" but only that it has survived refutation by experience better than its competitors, and therefore that the system "believes" it is true or right.

The Knowledge Base and Knowledge (Three)

- So instead of knowledge as "true, justified belief," the position taken here is that knowledge equals justified belief that some conceptual formulation, fact, or evaluation, is true or right as the case may be.

- In a very real sense, a system's knowledge is ***the analytical network of propositions and models constituting the knowledge base.*** It is therefore, just for convenience, that one may refer to a particular proposition or model as something a system "knows," because it knows that "something," only if one assumes that numerous unspecified background propositions and models are also known by it.

The Knowledge Management Process and Knowledge Management

- The Knowledge Management Process (KMP) is an on-going persistent interaction among human-based agents who aim at integrating all of the various agents, components, and activities of the knowledge management system into a planned, directed process producing, maintaining and enhancing the knowledge base of the KMS.

- Knowledge Management is the human activity within the KMP aimed at creating and maintaining this integration, and its associated planned, directed process.

The Knowledge Management Process and Knowledge Management (Two)

- A good way to look at the human activity called knowledge management is through the concept of the Use Case. In a use case a human-based agent, within the KMS, called an actor, participates in the KMP to get an outcome from the KMS that has value for the actor. The KMP can be represented as a set of Business Process Use Cases each classified within one of four business sub-process categories: planning, acting, monitoring, and evaluating. *A way of decomposing knowledge management activity then, is in terms of the use cases that constitute it*.
- The set of all use cases aimed at creating and maintaining the integrated, planned, directed process producing, enhancing and maintaining the KMS knowledge base, is an alternative characterization of knowledge management. The set of these use cases represents all of the organizational knowledge management activity of the actors making use of the KMS through the KMP. In other words, the set of use cases is what we mean by knowledge management in a human system.

Data, Information, Knowledge, and Wisdom

- What is the difference between data, information, knowledge, and wisdom?
- To begin with, organizational data, information, knowledge, and wisdom, 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, knowledge, or wisdom. Instead, to use a word that used to be more popular in discourse than it is at present, we are trying to specify intersubjective 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? Yes, information is provided by a datum, or by data, but only 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 which is a member of that class, some ideas about object operations or behavior, and relationships to other objects and classes.

Data, Information, Knowledge, and Wisdom (Two)

- 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.
- Wisdom, lastly, has a more active component than data, information, or knowledge. It is the application of knowledge expressed in principles to arrive at prudent, sagacious decisions about conflict situations.

Organizational Knowledge

- Organizational 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 (the organization’s version of objective knowledge).

- From the viewpoint of the definition given of organizational knowledge, what is an organization doing when it validates information to produce knowledge? 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.