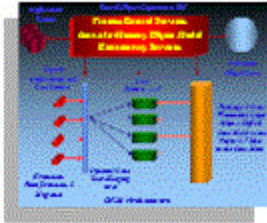
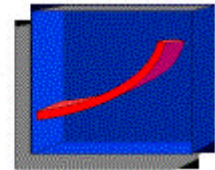


*EIS*

## DKMS Briefs

*Joseph M. Firestone*

### DKMS Brief No. Four: Business Process Engines in Distributed Knowledge Management Systems

#### Business Process Engines

John Rymer, in a White Paper written for Persistence Corp. recently defined "a new category of application server software" called the Business Process Engine (BPE).

"Business Process Engines manage the most important business state both in a fast in-memory environment and in close coordination with back-end databases. Business Process Engines process many user requests without help from a database." [1] In addition, they specialize in complex business rule processing, because their ability to maintain state is a special advantage in performing such processing.

"Business state is the information that describes the momentary status of the organization. To create business state, most applications acquire data from a database and then load it into memory for manipulations by the user." [2]

This is the "stateless" approach because, in it, a back-end database manages state. A key advantage of BPEs over stateless application servers, and what makes the BPE an important category to distinguish, is its ability to support scalable performance in distributed processing applications. Stateless servers must continuously recreate state in serial interaction with back-end databases. This consumes information bandwidth and is a major cause of slow performance in such systems.

By maintaining state in memory, BPEs greatly reduce the need for immediate and continuous requests to back-end databases, during business rule processing. They therefore reduce the workload and increase the performance of such data stores, and in addition, reduce network load and maintain the availability of bandwidth, thus allowing greater scalability of number of users and their requests for knowledge and information from a distributed system.

BPEs are also important for another reason not covered by Rymer. They represent key physical components in Distributed Knowledge Management Systems (DKMSs) and in DKM architecture. I will return to this role later, when I discuss the relationship of BPEs to DKMSs.

## Application Servers and Business Process Engines

Application servers provide services to other components in a distributed processing system by executing business logic (business rules) and data logic on data accessed from database servers. The class of application servers is subdivided by Rymer's distinction between "stateless" and in-memory server environments. Among stateless servers he distinguishes: Web Information Servers (they provide access to databases from web browsers and are alternatives to the CGI interface); Component Servers (they "provide data access and interaction frameworks for software components"); and Transaction Processing Monitors (they coordinate transactions within a distributed system).

Among Business Process Engines he does not provide a formal characterization of types of BPEs, but discusses some differences among Persistence's PowerTier product, which he considers the first Business Process Engine, and products from Gemstone and Angara that share some characteristics of BPEs, but don't quite fit his concept. Rymer does treat in some detail the distinction between "stateless" application servers and BPEs. Table One provides a comparison, closely-based on Rymer's [3], but somewhat abstracted from his comparison of three application servers, and is also useful in elaborating on the definition of a BPE.

Table One -- Business Process Engines and Stateless Application Servers

<b>Features</b>	<b>Business Process Engines</b>	<b>Stateless App Servers</b>
Client/server and Internet Support	Yes	Sometimes
Parallel Application Processing	Yes	No
Data-Driven Event Notification	Yes	No
In-Memory Application Model	Yes	No
Server-based Rules	Special Processing Capability	Yes
Transaction Isolation/Recovery	Partial	Partial
Business State Management and Synchronization	Yes	No
Proactive Monitoring	Yes	No
User State Management	Yes	Yes
Database Connection Pool	Yes	Yes

## Distributed Knowledge Management Systems and Business Process Engines

In previous White Papers and DKMS Briefs [4] I defined the DKMS concept and described an accompanying DKM architecture. The Active Knowledge Manager is a key logical component of this architecture. The AKM, a logical component of the DKMS, provides process/control distribution services, an in-memory active object model accompanied by a persistent object store, and connectivity to a variety of data stores and application types.

The AKM in DKM architecture, is composed of a set of distributed physical components that collectively provide the process control services, object model and connectivity required. These components, in turn, are BPEs sharing a set of reflexive objects [5] resident on the BPEs, and constituting a shared distributed in-memory object model. A reflexive object is one that is aware of changes in its state. Indeed, an AKM may be viewed as a distributed BPE that manages and integrates multiple processes through its BPE components.

In providing host platforms for the DKM shared objects, BPEs provide the essential integrative "glue" that holds the DKMS together. It is the ability of BPEs to maintain the shared object model in-memory and to manage and synchronize changes in it, that is the foundation for DKMS evolution over time, and provides the essence of its integration. This integrative role of the BPE is even more important than its role in enhancing performance.

But BPEs are also important, as Rymer says, because they improve transactional performance among the components of complex distributed systems, whether or not the primary application focus of the DKMS is in the area of OLTP or DSS processing. In both OLTP- and DSS-oriented DKMSs, good transactional performance among application servers is hard to achieve because of the increasing variety and complexity of the distributed components the DKMS must manage. Without the in-memory management of business state provided by BPEs, adequate performance in DKMSs could not become a reality.

BPEs hosting shared objects may support a variety of types of processes and applications in a DKMS. Persistence Software provides a good illustration of a BPE in describing its PowerTier Server product [6] developed for managing stock transactions. But there are many BPE types, and I will discuss a classification for them now.

### **Types of Business Process Engines and Tools for Implementing Them**

Here are some criteria for defining types of Business Process Engines:

- whether they are distributed across physical components or not;
- whether a BPE application server deals with a single or multiple business processes;
- the business process the BPE supports.

Clearly, distributed BPEs can be a powerful tool for upgrading performance in DKMSs, as well as for integrating their various components. BPE construction tools differ in their ability to support BPEs. Two construction tools that do support distributed BPEs are Daman Consulting's InfoManager [7] and Template Software's Enterprise Integration Template (EIT) [8].

An AKM is just a BPE that is both distributed and encompasses all of a DKMS's processes. A multi-process BPE can fall short of being an AKM, and instead can be restricted to a cluster of related processes. So, there are at least three types suggested by this criterion: a single process BPE, a BPE cluster, and an AKM.

How well a multi-process BPE performs will be correlated to the extent of its distribution, and to the complexity of the process it must support. But holding complexity constant, single process, non-distributed BPEs will generally perform better than multi-process non-distributed BPEs. So, multi-process BPEs will generally be distributed BPEs.

The third criterion for classifying BPEs is the business process supported. Persistence's PowerTier, once again, supports stock trading. Kevin Dick [9] mentions Financial Risk Management, Telecommunications Service Provisioning, and Transportation Scheduling as other process-driven applications. Here is a classification of BPE application servers based on Knowledge Management (KM) and Data Warehousing sub-processes.

- Planning

A planning BPE is a distributed BPE supporting a collaborative planning work flow. A tool for developing collaborative work flows including planning processes is Template Software's Work Flow Template. The Work Flow Template(WFT) is a specification of Template's SNAP 8.0 Foundation Template, and O-O Development Environment that produces Distributed Business Process Engines. WFT implements a Work Flow application framework on top of SNAP.

- Extraction, Transformation, Migration, and Loading (ETML)

An ETML BPE is a single component BPE performing data extraction, transformation, migration, and loading functions, and increasingly, data cleansing. Current ETML tools don't maintain state in an active in-memory object model and therefore don't qualify as BPEs. Instead, state is maintained in a dynamic data staging area and in a metadata repository. To make the jump to a BPE, current tools would have to incorporate an in-memory object model into ETML applications alongside the two data stores. The in-memory object model would improve the transactional performance of ETML applications in adding to or updating data in the DKMS

- Knowledge Discovery in Databases (KDD) along with Knowledge Base Modification, and Application (KBMA)

This is another distributed BPE supporting collaboration among investigators engaged in KDD and KBMA (a process of elaborating the knowledge base through deduction of new knowledge and mapping generalized knowledge to specific cases). KDD and KBMA must support messaging among knowledge workers in collaborations, work flow, evaluation of models according to complex validation rules, and other activities included in KDD and data mining and involving complex business rule-based computing. [10]

- Knowledge base/object/component model maintenance and change management

This process-driven application involves the AKM itself along with its dynamic integration capability. [11] Its distributed BPE supports monitoring of all changes in the Active in-memory object model, notifying other objects and components of such changes, and largely automatic adjustments of all objects and components to such changes. The processing associated with the AKM can be complex, and may incorporate intelligent agents. It certainly incorporates complex transactions in adjusting data warehouses and data marts to one another [12]

- Knowledge Publication and Delivery (KPD) to knowledge managers and consumers

The KPD application server keeps profiles of knowledge consumers and then interacts with them to decide on their specifications for knowledge consumption and delivery, including security specifications. Based on these specifications, which become business rules guiding publication and distribution decisions, this BPE publishes and delivers knowledge objects to consumers. Such objects can be documents, substantive models, software, or any of the other components of the corporate knowledge base.

- Computer-Based Training (CBT)

BPEs can also support re-skilling of employees in corporations and other organizations. Problem-based practice can now be integrated with automated instruction in a BPE where "the performance of the trainee is dynamically assessed against scenario specific expectations and performance standards." [13] This BPE uses cognitive modeling techniques to solve realistic simulation scenarios, and "to generate dynamic and adaptive expectations of future trainee actions." [14]

- Query and Report creation, modification, scheduling, and delivery

As reporting tools have evolved, they have begun to migrate to the middle tier in the form of Query and Reporting Servers. [15] Such servers mediate between client or web-based reporting tools and back-end databases both relational and multidimensional OLAP. These BPEs search, sort, and manipulate data. They can execute complex business rule processing. They remove substantial processing load from the desktop. In addition, they perform batch scheduling and production reporting. Reports may be created on the client and then executed on the BPE server. They can be scheduled for specific temporal intervals according to custom-designed reporting schedules. A library of canned queries and reports can be maintained on a server repository, and can be made available to any client.

- Relational On-Line Analytical Processing (ROLAP).

ROLAP Servers have been in use in the data warehousing field for some time [16]. Their function is to mediate between querying and reporting tools and back-end data warehouses and data marts. They employ complex business rule processing to support aggregations, rankings, and calculations not supported or well-supported by back-end databases. The support agents and alerts, multi-pass SQL, denormalization, specialized query processing and scheduling, and time series analysis. They also perform caching in order to upgrade performance, and to provide support for certain types of relational data models such as star and snowflake schema, that may not be supported by a database.

Of course, the three criteria for classifying BPEs, and the various types I've just distinguished are far from providing an exhaustive classification. But hopefully they provide some idea of the possibilities inherent in BPE development.

---

## References

[1] 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, P. 1, at [http://www.persistence.com/products/wp\\_rymer.htm](http://www.persistence.com/products/wp_rymer.htm)

[2] Ibid.

[3] Ibid. P. 9

[4] I introduced the DKMS concept in two previous White Papers "Object-Oriented Data Warehouse," and "Distributed Knowledge Management Systems: The Next Wave in DSS." DKM architecture was introduced in a third White Paper "Architectural Evolution in Data Warehousing." See also, "DKMS Brief No. One: The Corporate Information Factory or the Corporate Knowledge Factory?" and "DKMS Brief No. Three: Software Agents in Distributed Knowledge Management Systems." All are available at [http://www.dkms.com/White\\_Papers.htm](http://www.dkms.com/White_Papers.htm).

[5] Reflexive objects are used in Template Software Enterprise Integration Template product. See Template Software, "Integration Solutions for the Real-Time Enterprise: EIT - Enterprise Integration Template," Dulles, VA, White Paper May 8, 1998, P. 17 at <http://www.template.com>.

[6] Persistence Software, "The PowerTier Server: A Technical Overview" at [http://www.persistence.com/products/tech\\_overview.html](http://www.persistence.com/products/tech_overview.html)

[7] Daman Consulting is at <http://www.damanconsulting.com>.

[8] Template Software, Op. Cit.

[9] Kevin Dick "Sophisticated, Scalable Web Applications: Taming the Beast" at [http://www.persistence.com/products/wp\\_white.html](http://www.persistence.com/products/wp_white.html).

[10] See the description of the Perform Knowledge Discovery in Databases Use Case in my "Knowledge Management Metrics Development: A Technical Approach" White Paper No. Ten at [http://www.dkms.com/White\\_Papers.htm](http://www.dkms.com/White_Papers.htm).

[11] I've discussed dynamic integration in three papers: "Architectural Evolution in Data Warehousing," "DKMS Brief No. One: The Corporate Information Factory or the Corporate Knowledge Factory?" and "DKMS Brief No. Three: Software Agents in Distributed Knowledge Management Systems." All are available at [http://www.dkms.com/White\\_Papers.htm](http://www.dkms.com/White_Papers.htm).

[12] See Ibid. "Architectural Evolution . . ." and "Data Warehouses and Data Marts: A Dynamic View." Also at [http://www.dkms.com/White\\_Papers.htm](http://www.dkms.com/White_Papers.htm).

[13] The technology supporting this BPE is outlined in W. Zachary, J. Ryder, J. Hicinbothom, and K. Bracken, "The Use of Executable Cognitive models in Simulation-based Intelligent Embedded Training." Quotes are from the Abstract. Available at the CHI Systems Inc. web site at <http://www.chiinc.com/ginapubs.html>.

[14] Ibid.

[15] See IQ/Smart Server from IQ Software, recently purchased by Information Advantage, Inc. at <http://www.iqsc.com/products/iqss/iqss.htm>.

[16] The two leading products are MicroStrategy's DSS Server at <http://www.strategy.com>, and Information Advantage's DecisionSuite Server at <http://www.infoadvan.com>

---

## 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](mailto:eisai@home.com).

- 
- [ [Up](#) ] [ [KMBenefitEstimation.PDF](#) ] [ [MethodologyKIv1n2.pdf](#) ] [ [EKPwtawtdKI11.pdf](#) ]
    - [ [KMFAMrev1.PDF](#) ] [ [EKPeussol1.PDF](#) ] [ [The EKP Revisited](#) ]
    - [ [Information on "Approaching Enterprise Information Portals"](#) ]
    - [ [Benefits of Enterprise Information Portals and Corporate Goals](#) ]
      - [ [Defining the Enterprise Information Portal](#) ]
    - [ [Enterprise Integration, Data Federation And The DKMS: A Commentary](#) ]
      - [ [Enterprise Information Portals and Enterprise Knowledge Portals](#) ]
  - [ [The Metaprise, The AKMS, and The EKP](#) ] [ [The KBMS and Knowledge Warehouse](#) ]
    - [ [The AKM Standard](#) ]
  - [ [Business Process Engines in Distributed Knowledge Management Systems](#) ]
    - [ [Software Agents in Distributed Knowledge Management Systems](#) ]
    - [ [Prophecy: META Group and the Future of Knowledge Management](#) ]
      - [ [Accelerating Innovation and KM Impact](#) ]
      - [ [Enterprise Knowledge Management Modeling and the DKMS](#) ]
        - [ [Knowledge Management Metrics Development](#) ]
        - [ [Basic Concepts of Knowledge Management](#) ]
  - [ [Distributed Knowledge Management Systems \(DKMS\): The Next Wave in DSS](#) ]