

Available online at <http://www.mecspress.net/ijem>

## A Service-oriented Dual-bus BAM System Model

Yalei Yang <sup>a,\*</sup>, Xingshe Zhou <sup>a</sup>, Wenjian Xu <sup>a</sup>, Gang Yang <sup>a</sup>

<sup>a</sup> School of Computer, Northwestern Polytechnical University, Xi'an, P.R. China

---

### Abstract

Based on a thorough analysis and comparison of several business activity monitoring(BAM) system models, noting that a BAM system is open, integrative, and real-time, and using the SOA software engineering techniques, this paper presents a service-oriented dual-bus BAM system model and analyzes in detail the functions of its service components and communications among them. Much more open than prior models, this BAM system model has a two-level bus architecture, which enables flexible and extensible implementations of practical systems, as well as provides a foundation for software reuse.

**Index Terms:** Service-oriented; business activity monitoring; service components; application components; enterprise integration

© 2012 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science.

---

### 1. Introduction

Nowadays enterprises are faced with great pressures. It is important for the enterprise to adapt to the ever-changing market and construct a highly reactive environment in order to survive in the fierce competition. Although computer software products have already spread over many industries, enterprise information administration and investigation still depends on users manually triggering data management and function analysis. This kind of information always comes with delays and fails to figure out problems promptly, resulting in loss of opportunities and profits. Under the environment of a complicated and ever-changing market, enterprises exhibit heavy demands for real-time, open, highly-integrative and predictable analysis to win the competition. Business activity monitoring (BAM) system aims to meet such demands. The term BAM is invented by Gartner Group in 2002[1], and it is a software technique to monitor enterprise running status based on enterprise application integration.

This paper analyzes in a systematic way the data-centric, execution-centric, and object-oriented BAM system model structure as well as its advantages and disadvantages. Then we present an extensible dual-bus

\* Corresponding author.

E-mail address: yangyalei1983@163.com

model (DBBAM), which has clarified structure levels, standard module interfaces and simplified module connections.

## 2. BAM system model types

### 2.1. BAM system model goal

At present, the design goal of BAM system model is to increase integration and extensibility of the BAM system in use. As a monitoring system, the ability of real-time response is considered as well.

The openness of BAM system, which is decided by the system model, includes its data openness, function openness and system extensibility. The integration of BAM system, which is also decided by the system model, refers to integrating enterprise applications in use into an integrative environment by coherent description information. The real-time feature of BAM system is implementation related, and it refers to the efficiency of retrieving data from business activities and processing them.

### 2.2. Data-centric system model

A system model that is data-centric is shown in Fig. 1. Models of this type share the database in the system primary level; all functional components use unified data descriptions; all sub-systems are independent; the system in whole so extensible as to add at will applications that conform to its data exchange standards. The disadvantage is that this system model is not real-time when presenting information to users, because it writes data to database and BAM reads data from database before processing them, which introduces delays. Relevant research works include: Jin Gu Kang et al.[3] designed a business performance management system according to the business activity monitoring requirements of a multinational car corporation.

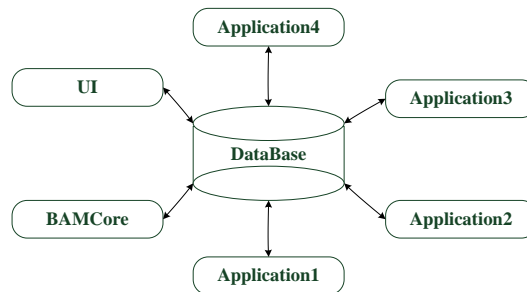


Fig.1 Data-centric system model

### 2.3. Execution-centric system model

A system model that is execution-centric is shown in Fig. 2. Models of this type integrate different enterprise applications by BAM execution center; mutual processes of data(data extracting, data transforming, and data transporting) are separated from applications and placed in execution center to reduce redundancies; all database access operations are carried out by the execution center to achieve strict data management and insure data consistency. The disadvantage is that the execution center in this model is too complicated in function design because it not only communicates with both user interface and applications, but also manages BAM database at the same time. The burdens are overwhelming and apt bottlenecks are introduced. Relevant research works include: Josef Schiefer et al.[6] studied business activity monitoring data storage and presented one kind of storage technique (PDS) that supports real-time data processing.

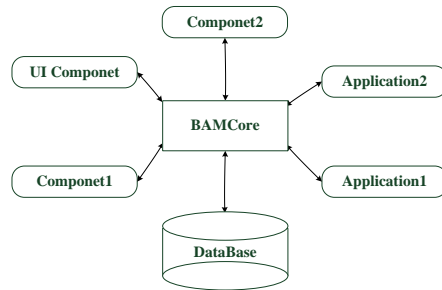


Fig. 2 Execution-centric system model

#### 2.4. Object-oriented system model

With the maturity of object-oriented techniques, more simplified object-oriented system models emerge as shown in Fig. 3. BAM core objects encapsulate data and methods shared by all user interface objects, application objects and service component objects; user interface objects encapsulate user interface data and relevant methods; application objects encapsulate application data and relevant methods; service component objects encapsulate methods that provide certain service for BAM. All these objects accomplish the desired tasks through mutual communication and cooperation. The advantages of object-oriented model include: proper encapsulations of data and methods reduce the communication overhead and operation complexity of centralized management. References [7][8] adopt the object-oriented design pattern to build their BAM system models.

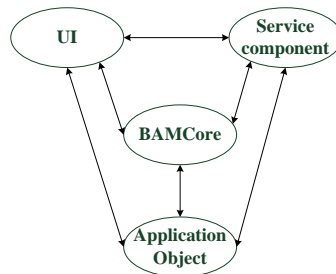


Fig. 3 object-oriented system model

#### 2.5. Service-oriented system model

Enterprise software systems become more and more huge and complex, and functions of systems are repeatedly developed, resulting in wastes of resources and inconvenience of management. SOA is such a software architecture with independent function entities that resolves enterprise application integration and accomplishes special tasks through links. SOA is a component model, and it interconnects different service components of applications through communication protocols. SOA-based system architecture has the following features[9]:

- 1) Loose coupled: the integration of applications is through enterprise service buses, and modifications of one part do not affect other parts.
- 2) Inter-operative: based on protocols, inter-operative solutions of multi vendors are provided, which can be implemented by several techniques.

3) Position transparency: SOA provides position transparency through its “publish/subscribe” mechanism. Service consumers need not know the physical positions of service providers.

SOA-based structure and BAM system model [10][11] is shown in Fig. 4: enterprise applications are integrated through enterprise service buses, and service components deal with data collection, execution, and presentation. The trick of this model is that it has an efficient bus structure which enables interconnections among its components by common interfaces. Components are seamlessly integrated and are plug-and-play. Communication links among components in this system model are linear and have unified interfaces and are thus less complicated and more cooperative.

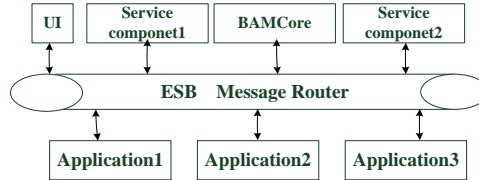


Fig. 4 Service-oriented system model

However, with the increase of application components, the bus will be burdened with more and more requests, which introduce delays and affect the real-time ability of the BAM system. To solve the problem, this paper presents a service-oriented dual-bus system model, shown in Fig. 5. The second bus supports specially application components, maintaining fine interoperations between application components and primary components.

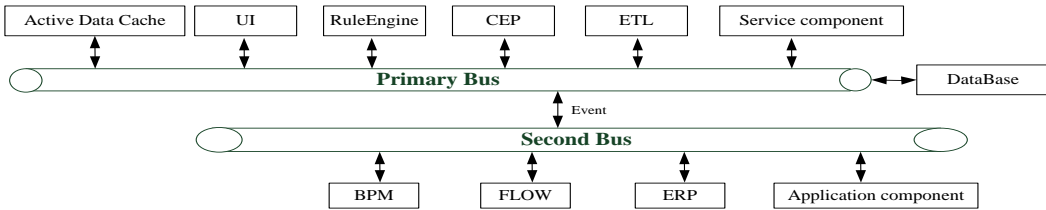


Fig. 5 DDBAM system model

### 3. DDBAM System model infrastructure

#### 3.1. Bus

The bus provides communication transparency mechanism among components. It exposes unified interface standard, requiring every newly added component to register its function service information and deregistering itself before exiting. When one component requests the service of another, the bus searches the function service information library, locates the component that provides the service, and transmits the service request.

#### 3.2. Primary components

The primary components in this model are as follows: complicated event processor, ETL, active data cache, database, rule engine and user interface.

- 1) Complicated event processor(CEP)

CEP acts as the event processing engine of the DDBAM system. When enterprise application components fire events and change status during execution, it will encapsulate them into atom events through the adapter and hand them over to the event handler engines. CEP mainly provides functions such as event filtration, event composition and event consultation. The event filter discards of irrelevant events that are of no DDBAM interests; the event composer performs effective compositions of atom events and compound events according to business rules; the event consulter uses structured query language (SQL) to search events in CEP.

### 2) ETL

ETL is the data processing engine of DDBAM. It performs data extracting, transforming and loading. ETL data extraction is different from the traditional data exaction way. The traditional way is to extract the desired data from the data source and do some cleaning to the data and finally load them to data warehouse according to predefined models, whereas in the DDBAM system, data are extracted from event flows and transformed into enterprise business metrics and finally loaded into the active data cache.

### 3) Active data cache and database

The active data cache temporarily stores active data of the current business of the DDBAM system and the database permanently stores them. BAM differs from traditional business process monitoring in that it provides real-time monitoring. The DDBAM writes data to the active data cache first and then writes them permanently to the database on a regular basis.

### 4) Rule engine

The rule engine is the business logic processing engine of the DDBAM. It separates enterprise business logics from hard code and stores them in the rule library which loads the business data from the active data cache to the working memory and matches them via fast matching algorithms. When the user-defined rule matches the current business data, alarm will be activated automatically, and one email or short message will be sent to the administrator. The rule engine brings flexibilities to the DDBAM system. Modifications to rule files will cater to enterprise business logic changes, leaving source code untouched.

### 5) User interface

The user interface is comprised of the configuration view and the presentation view. The configuration view is used to configure the monitor model such as event model, business metrics model, key performance indicators and rule engines, etc. The presentation view displays the running status of the key performance indicators through intuitionistic diagram components, which include the dashboard, the histogram, the graph and the biscuit cart.

## 3.3. Application component

Application components fit for any DDBAM application areas such as ERP, Flow and BPM. The integration approach of application components and the DDBAM core system is shown in Fig. 6. Application components are connected to the bus through adapters which wrap the complicated communication process so that the application component can focus on business functions. Moreover, it dispatches data from the application component through events.

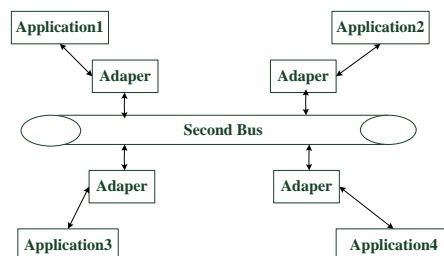


Fig. 6 The integration architecture of application components

### *3.4. System openness support of service-oriented architecture*

The service-oriented architecture derives and extends the object-oriented architecture, and it has all its advantages. Furthermore, a service component is a function component that is well encapsulated and has unified external interfaces. Communications among components are wrapped in the bus, decreasing dependencies of service components. Software developers can conveniently define and construct new plug-and-play components to extend the system. The DDBAM system model's employment of service components increases enormously the openness and extensibility of the system.

## **4. CONCLUSION**

With the expansion of enterprise scales, applications will become more and more complex. In order to survive in the fierce market competition, enterprises need to be equipped with the required capabilities of an agile enterprise: to quickly responds to changes of the market, to quickly adjust business strategies, and to quickly deploy business workflows. The business activity monitoring (BAM) system shortens the data processing time of enterprise applications and makes fast optimizations and adjustments to business workflows according to business rules through the real-time information of the status and results of multiple business operations, business processes and business transactions. Yet how to construct an effective BAM system upon existed enterprise applications with fewer burdens of application components and good system openness and extensibility still awaits further exploration.

## **Acknowledgements**

This work is sponsored by China National High Technology Research and Development Program (863 Program) under Grant No. 2009AA010308, and Major national science and technology projects-- series of middleware development and production industrialization under Grant No. 2009ZX01043-002-001.

## **References**

- [1].D. W. McCoy. Business activity monitoring: Calm before the storm[R]. Technical Report LE-15-9727, Gartner, April 2002.
- [2].Wikipedia, "Business Activity Monitoring",[http://en.wikipedia.org/wiki/Business\\_activity\\_monitoring](http://en.wikipedia.org/wiki/Business_activity_monitoring), 2008.
- [3].J. G. Kang and K. H. Han, "A business activity monitoring system supporting real-time business performance management," in 3rd International Conference on Convergence and Hybrid Information Technology (ICCIT ' 08), vol. 1, Nov. 2008, pp. 473 - 478.
- [4].Savitha Srinivasan, Vikas Krishna, and Scott Holmes. Web-log-driven business activity monitoring. IEEE Computer, 38(3), 2005. <http://dx.doi.org/10.1109/MC.2005.109>.
- [5].J. G. Kang and K. H. Han, "A business activity monitoring system supporting real-time business performance management,"in 3rd International Conference on Convergence and Hybrid Information Technology 2008
- [6].Josef Schiefer, Beate List, Robert M. Bruckner. Process Data Store: A Real-Time Data Store for Monitoring Business Processes[C]. DEXA 2003. Springer-Verlag Berlin Heidelberg 2003: p.760-770.
- [7].J. Jeng, D. Flaxer, and S. Kapoor. RuleBAM: A Rule-Based Framework for Business Activity Management. In IEEE SCC, pages 262-270. IEEE Computer Society, 2004.
- [8].H. Kim, Y-H. Lee, H. Yim, and N. W. Cho, "Design and Implementation of a Personalized Business

Activity Monitoring System”, Lecture Notes in Computer Science, vol. 4553, pp. 581-590, 2007.

[9].Brown, A; Johnston, S.; and Kelly, K. Using Service-Oriented Architecture and Component-Based Development to Build Web Service Applications. Rational Software Corporation. 2002.

[10].Jiří Kolář. Business Activity Monitoring, MASARYK UNIVERSITY FACULTY OF INFORMATICS [J], Spring 2009.

[11].Tom Lubinski, SL Corporation, Corte Madera. Business Activity Monitoring: Process Control For the Enterprise[J].2008