A Data Secure Solution for Heterogeneous Enterprise Legacy Systems Integration

Bin Xu
College of Computer Science & Information Engineering, Zhejiang Gongshang University, Hangzhou, China
Email: xubin@mail.zjgsu.edu.cn

***Abstract*—To stay competitive, enterprises have to meet or even exceed the expectations of consumers. Business data integration can be an efficient integration method for the enterprises with several distributed departments and heterogeneous legacy business systems. Due to the different functionalities of the different business departments, the access authority to the business data is different. This paper suggests yet a new method to establish a security business data integration framework among heterogeneous legacy business systems.**

***Index Terms*—heterogeneous legacy system integration, data security, data integration**

I. INTRODUCTION

While the enterprises are now enjoying the suitable legacy sub-systems which can handle the functionalities for individual departments very well, they are obliged to integrate the business information among the entire enterprise due to the surprising improvement of information and communication technology and the increased business needs. However, those legacy sub-systems may be built with different programming language, different hardware infrastructures, different computing platform, different system architecture and different database to store the business data and hard to be integrated.

To stay competitive, enterprises have to meet or even exceed the expectations of consumers. While their core business activities are relaying on the legacy applications, the business data is isolated between these applications. In order to make the business run smoothly and efficiently, enterprises should evolve their legacy applications and integrate these applications seamlessly.

Business intelligence systems can help enterprises have more comprehensive knowledge of the factors affecting their business, such as statistics on sales, production, internal operations. In order to make better decision so as to survive in the competitive market, the companies are suffering in building an enterprise business intelligent system: to collect and analyze the business information dynamically so as to monitor the market and adjust the direction according to the demanding of market.

Manuscript received January 18, 2009; revised February 20, 2009; accepted March 20, 2009.

Corresponding author: Bin Xu, Email: xubin@mail.zjgsu.edu.cn

In short, there are two goals the enterprises are to achieve: one is to evolve their legacy applications and integrate them seamlessly so as to meet or even exceed the expectations of consumers, and another is to generate the applications into an enterprise business intelligent system so as to make better decision making. There can be several solutions to achieve these goals.

- *All-in-one solution*. Build a new enterprise business intelligent system, which can fulfill all the functions of the legacy business applications.
- *Self-evolving and light level integration.* Evolve the individual applications so that they can be easily integrated. The enterprise business intelligent system should also been developed.
- *Heavy-level integration.* Keep the legacy application unchanged. Build a new enterprise intelligent system to fetch all the business data from the individual applications.

These solutions vary from risk, cost, schedule, long term benefit, precondition and real-time response [Table 1].

TABLE I.
COMPARISON OF THE SOLUTIONS

Item	All-in-one	Light-level integration	Heavy-level integration
Cost of development	Large	Tiny	Medium
Cost of training	Large	Some	Tiny
Risk in development	Large	Some	Medium
Schedule	Long	Little	Medium
Benefit to Business (long term)	Efficient	Suitable & a little efficient	Suitable
Precondition	business logic is clarified	Internal Data flow is clarified	Data structure is understood
Real-time response	Yes	Yes	No

As stated, the all-in-one solution will benefit the enterprise in long term but with large risk, cost and last long in the development phase and at the initial phase of deployment. There is also some limitation for these solutions. In order to development all-in-one system, the business logic should be clarified before the development. Most of the time, reverse engineering will be performed if some business logic inside the applications is not well documented. Data flow should be clarified if we want to choose the light-level integration. Otherwise, we are difficult to integration the data

seamlessly. The requirement is rather weak for heavy-level integration as only data structure should be understood. However, such solution cannot get the real-time response.

In this paper, we propose yet another solution between the light-level integration and heavy-level integration. Such solutions are expected to achieve:

- *Minimal risk, cost and schedule,*
- *Suitable long term benefit,*
- *Real time response,*

while no business logic or internal data flow should be clarified.

A framework is proposed to seamlessly integrate the individual business applications into enterprise business intelligent system.

Agent technology can be a good candidate to be adopted. Since there are several difficulties of communication between intelligent agents, there should be a common syntax, semantics and pragmatics. A variety of inter-process information exchange protocols, including the directly reply model, remote call model and message queue model, has been suggested [15].

The enterprise information system is essential to business operations and the data inside should be kept private and security. In this paper, the main contribution of the authors is the proposal of an agent-based security business data integration middleware for heterogeneous enterprise legacy systems.

The rest of this paper is organized as follows. Section II introduces the inter-system communication and Section III presents the framework. Section IV presents the business information convergence method; Section V demonstrates the experiment and its result. Related work is introduced in Section VI and Conclusion and discussion are stated in Section VII.

II. Inter-system Communication

Typically the legacy enterprise sub-systems depend much on the different network protocols, are not scalable and hard to be improved. Distributed Agent technology is used in our research to solve such issues to a certain extent. Distributed Agent, which has self-learn ability, can timely and initiatively respond to the events and situation in the networks and facilitate the synergy between the systems and the middle-wares. Figure 1 shows the agent-based inter-system communication framework. Those agents can be the middleware attached to the heterogeneous legacy business sub-systems. As shown in Figure 1, the Agents can handle (including the collection, filtering, aggregation, and analysis activities) multiple data sources and configure the processed data in the different data warehouses.

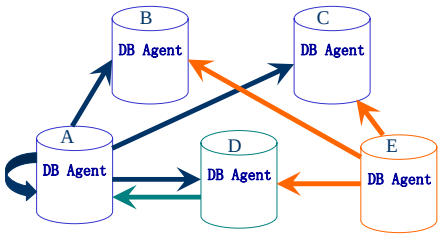

Figure 1. Agent-based inter-system communication

Figure 1 demonstrates three configuration modes, self-target, back-up, and distribution.

Self-target The agent at Node A can place the target data to Node A itself.

Back-up The agent at Node A and Node D can place their data to each other to backup some important business data.

Distribution The agent at Node E can distribute the target data to Node B, Node C, and Node D. Such distribution enables the parallel operating of transactions and improves the performance.

Some complicated events such as repeatedly trigger and multi-theme related trigger can be easily handled with such design.

III. Framework of Agent-based Middleware

A series of Agents, including the Agent dealing with the ECA rules, the communication Agent, the data aggregation Agent, the data display Agent and the Agent providing the database location service can be deployed in the middleware.

These Agents can then be deployed to different Nodes with the deployment of the middleware. They can monitor a specific event, issue preferred action, transfer the data and complete the transactions.

Figure 4 demonstrates the framework of agent-based middleware for enterprise legacy sub-systems. There are two parts in the middleware, while Part 1 acts as the local agents, Part 2 acts as the remote ones.

The events from the enterprise legacy systems, operating system, network system and the DBMS can be recognized by the event monitor agent. It can then contact the ECA (Event-Condition-Action) agent for sequent operation. ECA agent contains a configuration for the ECA rules. When it received event signal from event monitor agent, it will act according to the ECA rules. Some conditions may be checked from the enterprise legacy sub-systems. If the event turns out to be responsible, the operations will then be packed and transferred via the communication agent. Most of the time, Part 1 and Part 2 are not counterparts except the middleware is acting in self-target mode. So in Figure 4, the Part 2 refers to that of target middleware. When the communication agent received the operation package, it will unpack the package and send the commands to the data operating agent to handle the data operation. The data migration agent can tide up the database and balance the volume of active data and expired data. It runs in periods and can be issued by the data operating agent on demand. The information demonstration agent can integrate the data from the distributed databases and generate them into different views according to the needs from the managers.

A. ECA agent

ECA agent is used to handle the database operations according to the event and the event-condition-action rules. ECA mode has been suggested by Stanford University "WHPS" group as shown in Figure 2.

When we use such mode to deal with the business data integration among enterprise legacy systems, we found that the above mode is not scalable and sometime may not be security. For example, there is no interaction mechanism for the local DB to deliver the correct business data to the data warehouse. The monitor has to deal with the detail database operations, which reduce the entire performance and bring forth the data security issue.

In order to make the business data integration security and with high performance, we take the interaction mechanism into consideration. The data needed by the customer can be subscribed by the customer to a certain node. The authorization will be validated during the subscription and the communication.

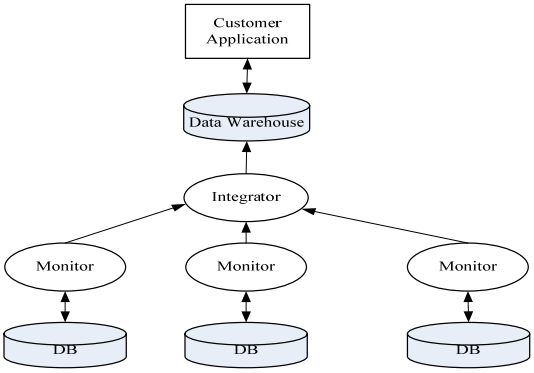

Figure 2. WHPS data monitoring mode

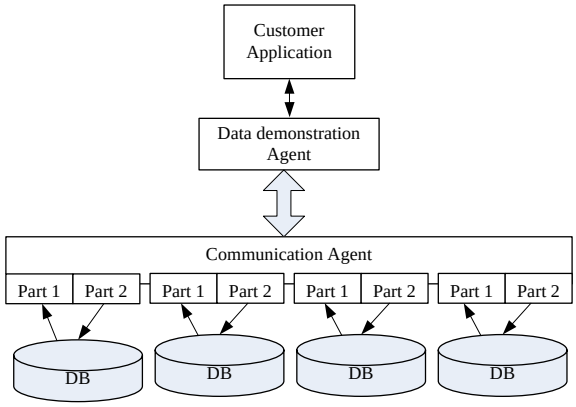

Figure 3. Agent-based logical data warehouse

In Figure 3, the customer can subscribe the needed business data via the middleware, the current available computing and storage resource become a logical data warehouse. While the performance has been improved and the exposure of business data can be controlled, the no additional computing resource is required.

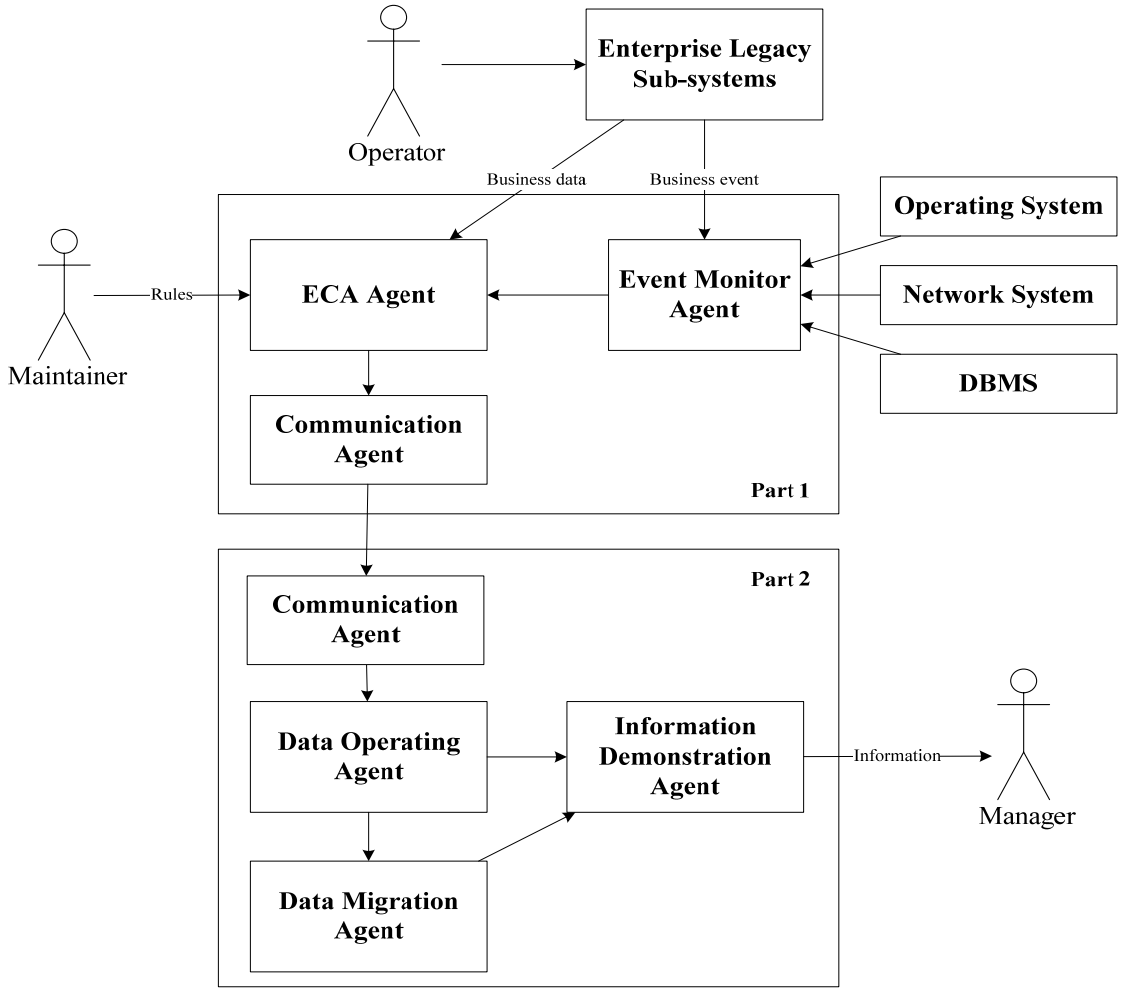

Figure 4. Framework of agent-based middleware for data integration

B. Communication Agent

There will be lots communication between multiple database servers and data storage server to transfer the operation package, in order to accomplish such tasks more security. A security communication agent is required.

According to TCP / IP communication two-way symmetry, we designed a Session class for the management server, the client need to address common data and operation. In addition, under the category Session link encryption, Agent tasks, provide encryption, decryption and implementation of remote operation services.

Class ClientAgent succeeds from Class Session to handle communications tasks, pegged TrigerAgent to realize the initiative data transfer in the logic data warehouse.

Class ServerSession succeeds from Class Session, in the node each ServerSession instance corresponds to a certain client, responsible for handling contacts communications.

Class ServerAgent monitors network port, generates and manages ServerSession instances for the client.

Class CipherAgent encrypt the data through the DES algorithm, and prepare the random encryption handshake validation. The hide Key of DES algorithm is generated at the server side and transferred through an absolute credible channel such as encrypted e-mail.

When a session received a remote package, it will convert the content into a Task instance, which will handle the consequent activity.

Class TrigerAgent connects the expansion stored procedure in the database through the monitoring of the network package. It will then recognize the event timely and initiatively through the Class TaskAgent and Class ClientAgent.

Data flow from the perspective of system operation can be divided into the following process steps.

1. ClientAgent landing ServerAgent
   a) ClientAgent establishs a connection with ServerAgent using TCP protocol.
   b) ClientAgent use of public key to send its own landing Key logo, test whether it has the right landing key.
   c) ServerAgent use of public key landing key customers read labels and then loading and search key customers.
   d) ServerAgent use CipherAgent randomly generated problems, and encryption key used to send customers to ClientAgent, to test whether they really have their own declared independence key.
   e) ClientAgent use their own server's decryption key issues, the certification of their own keys correct copy, and answering the question, the answer will be key encryption with their own answer ServerAgent.
   f) ServerAgent read the answer, decryption recognized each other after landing at the same time have the right key and correct its own key, and clearly the correct logo, landing process is complete.
   g) Through two sets of keys, as well as the question and answer random process, landing process avoids the use of a simple user name, password approach to landing brought about by the security issues, but also to prevent eavesdropping network provides an effective means.
2. Active Data Transmission
   a) Active Database issued UDP data packet to TrigerAgent.
   b) TrigerAgent connecting data warehouse initiative, which removed cache in the operation records, the process of using his father's encrypted communications functions ClientAgent these messages will be sent to the data warehouse server management ServerSession ServerAgent.
   c) SeverSession read these data, to TaskAgent processing.
   d) TaskAgent identify the type of data to test the limits of the other keys, generating SQLTask examples processing
   e) SQLTask linking data storage servers, the implementation of data operation.
   f) Figure 5 shows the relevant Agent relationship. Which DBServer said sources in the communications database, DATA WareHouse said that the objective logic of the data warehouse.

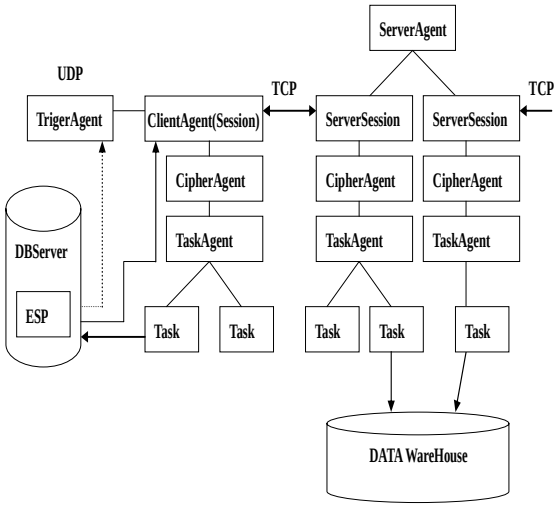

Figure 5. Structure of communication agent

IV. BUSINESS INFORMATION CONVERGENCE

Business information convergence includes business information definition, business information gathering, business information filtering and business information mergence and analysis. Here in this section, we will focus on how to establish the information transfer channel and how to transfer the information data among the individual business applications.

A. *Information transfer channel definition*

Business information can be transferred between applications and the transfer may invoke some other business information transfers. A business information transfer between heterogeneous business applications can be demonstrated as a hierarchical network.

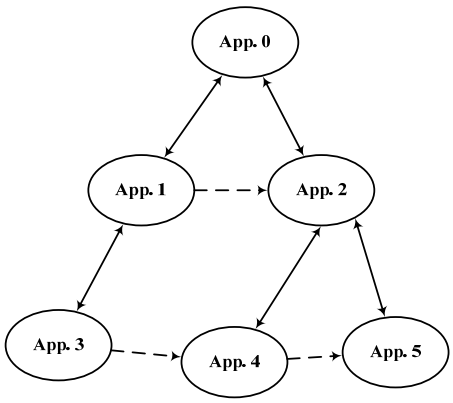

Figure 6. Example of business information transfer in a hierarchical network

Figure 6 is an example for the hierarchical organization of the business information transfers. While App. 0 transfer information to App. 1 and App. 3. The transfer to App.1 invokes the information transfer between App.2 to App.0. The transfer between App. 1 and App. 3 invokes the transfer between App. 4 and App. 2. Consequently, the information transfer between App. 5 and App. 2 will also be invoked.

An entire business information transfer and invoking can be run in following procedure:

{Driver chooses a channel of information transfer and put all these transfers into transfer stack.}

While {the transfer is not finished as shown in the transfer stack}

{

{Transfer the required information to the destination}

{Invoke the other information transfer as required:

put the invoked transfer into the transfer stack}

}

For instance, the information transfer and invoking in the example can be demonstrated as follows.

1) Put App.0->App.1 and App.1->App.3 to the stack.

2) Transfer the information from App.0 to App. 1

2.1) Put App.2->App.0 to the stack

3) Transfer the information from App. 1 to App. 3

3.1) Put App.4->App.2 to the stack

4) Transfer the information from App. 2 to App. 0

5) Transfer the information from App. 4 to App. 2

5.1) Put App.5->App.2 to the stack

6) Transfer the information from App. 5 to App. 2

The information transferred should also be defined in the channel and put in the stack during the transfer.

B. *Business Information Registration and Subscription*

As shown in Figure 7, each piece of the business information can be defined by a certain provider and can be subscripted by the consumer which can be either the other applications or the enterprise business intelligent system. To facilitate the definition and subscription, the business information may be cataloged and can be identified by keywords.

Three data structure should be defined, including Global Catalog Index, Registration Table and Subscription Table. Global Catalog Index used to unify the information catalog among all the heterogeneous units so as to reduce the possible semantic conflict. Registration table is used for the information providers to register the available information. The subscription table is used to record the consumer's information request.

Definition 1: Global Catalog Index

Global Catalog Index::= <Information Code><Business Information Catalog>, where information code can be the alphabet words.

Business Information Catalog ::= [Any information should be transferred between applications or to the enterprise business intelligent system]

Definition 2: Registration Table

Registration Table::= <Application> <Information Code>, where Node is the ID code for the provider node.

Definition 3: Subscription Table

Subscription Table::= <Application> <Information Code>, where Application is the identification code for the consumer code.

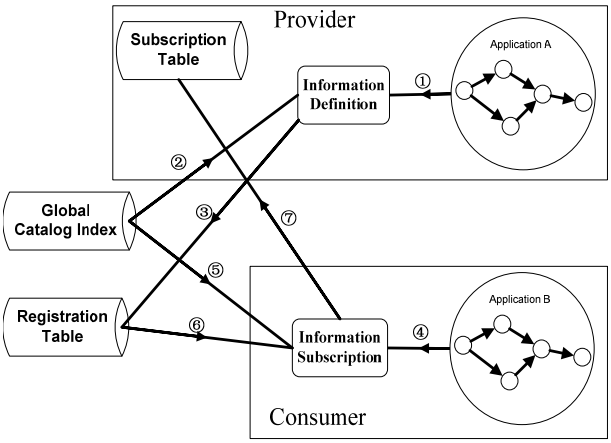

Figure 7. Business information identification interface demonstration

The process for the provider to register its information can be separated into three steps.

1) Provider identifies the information contained in local application.
2) Provider fetches the catalog index code for the information to be provided.
3) Provider registers the information according to the catalog described in the global catalog index.

At the other hand, the consumer subscribes the information in four steps.

4) Consumer receives the request of information from its local application.
5) Consumer fetches the catalog index code for the required information.
6) Consumer searches registration table to find out the suitable provider has such information.
7) Consumer makes the subscription with the certain provider.

C. Business Information Gathering, Transferring and Filtering

As soon as the registration and subscription of the information has been completed, the business information gathering, transferring and filtering can be started consequently.

As shown in Figure 8, the business information can be gathered by the provider from its local application in a certain period with pre-designed interface and mechanism. The mechanism includes tools, middleware and some manually operations. The gathered information can be put into a cache. The information transferring process serves as an agent, which transfer the cached information to the subscribers. The consumer filters the received information data on demanding and sends the final information to its local application.

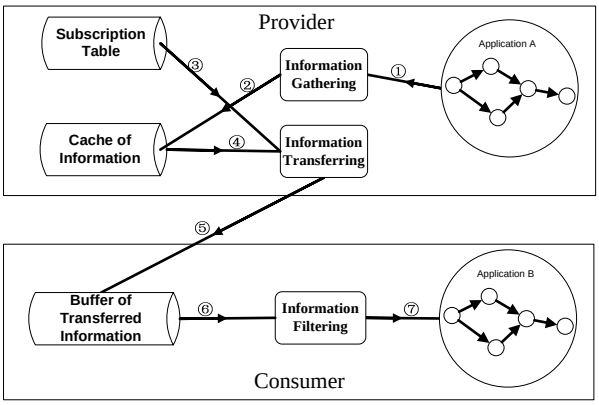

Figure 8. Business information gathering, transferring and filtering

The process for the provider to gather its information can be separated into two steps.

1) Provider gathers the information contained in its local application and formats the data contained in the information into unified catalog.
2) Provider caches the information in its cache.

The information transferring is handled by an agent in three steps.

3) The transfer agent is noticed by the event that there is new information has been cached.
4) The transfer agent checks the subscript table to get the location address.
5) The transfer agent packs the information and sends it to the destination.

The information filtering is handled by an agent in two steps.

6) The filter agent fetches the buffered information when it is noticed by the event that new information has been buffered.
7) The filter agent unpacks the information and localizes the data. The final information will then be fed to the destination's local application.

D. Information Flow

As shown in Figure 9, there will be five data process centers in the information flow, which can be implemented as agents. These agents can be driven by provider, consumer and the data from the storage, including the information cache and the transferred information buffer.

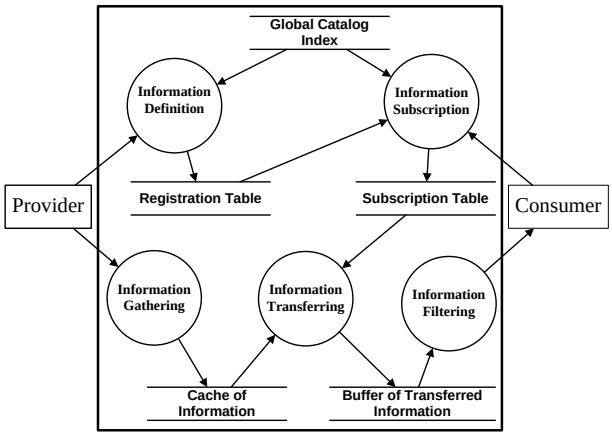

Figure 9. Related information flow

E. Intelligent Information Policy Making Framework

On the basis on the dynamic information retrieval process proposed in Section 3.1 and the information convergence method proposed in Section 3.2, intelligent information policy making framework can be established as Figure 10. The maintainer performs the tasks of policy making, and the other agent conduct the dynamic information retrieval.

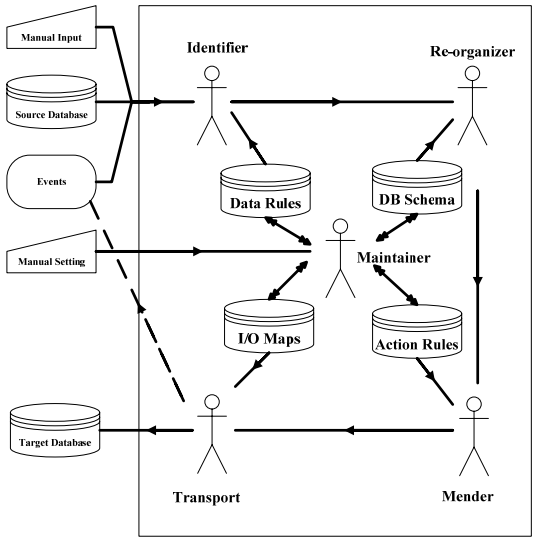

Figure 10. Intelligent information policy making

V. CASE STUDY

A traditional database system which was handling storage management for a large enterprise in Hangzhou, China was chosen for the experiment. This database system had been run for several years, and it undertook the daily storage management. Though the managers wanted to get some useful integrated metrics, they couldn't afford to halt the system for a long time. Therefore, to update the original database system is full of risk due to the possible software defects and the unknown business logic and data/control flow insides the system.

We set up the experimental environment and a simple demon [Figure 11] was built to show the features of our project.

There were two boxes containing traditional database system and the new added adaptive data warehouse, both of them form the adaptive information retrieval mechanism.

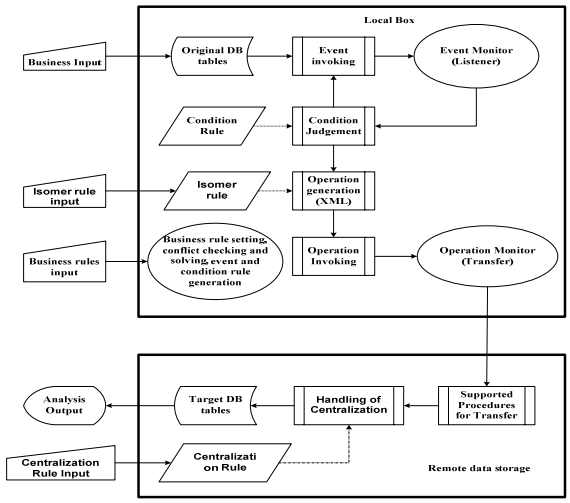

Figure 11. System architecture of the prototype

Some triggers were created for 2 tables inside the original database system. One Java application was installed in the box containing the original database system. This Java application monitored the events thrown from the original database system, and generated the information if certain conditions were satisfied. The generated information was then stored into the adaptive data warehouse for the further analysis.

The middleware was validated in 4 depots with separated legacy management information systems. Because those systems were developed by different vendors, they are using different DBMS: two of them are using Sybase Database, one is using MS Sqlserver, and the rest one is using Oracle. The codes in the systems are not unified and the team found that there will be several names and codes for the same objects. While at the same time, the same codes also refer to the different objects. We expand the ECA rules to deal with some simple code transfer and generate the business data integration rules. Figure 12 shows one data transfer definition.

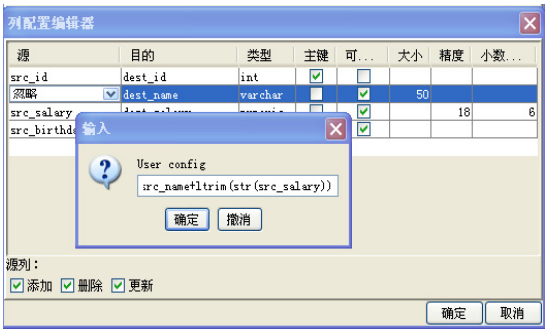

Figure 12. Data transfer definition

The team spent nearly one month to define the rules. We were fortunately to find the return was wonderful. The management can have some uniform reports and they don't need to make calculation manually. The working performance has been improved largely. Besides, the management valued the speed information demonstration very much.

Considering the business data security issue, here we only demonstrate some examples of the quick data reports.

Figure 13. Quick report example

VI. Related Work

When dealing with unstructured or semi-structured data, Information filtering and information retrieval techniques can be used. Information filtering has been defined as a variety of processes [3] involving the delivery of information to users [2]. It's an information seeking process in which documents are selected from a stream of incoming data to satisfy a relatively stable and specific information need [4].

Salton defined information retrieval in his classic textbook [1] as a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information. With the increase in the amount of online available text and the demand for different types of information access these years, the information retrieval related areas have been renewed and gone beyond simple document retrieval to summarization and multimedia retrieval (e.g., music, image and video).

However, traditional information retrieval process is driven by the query of user. Those companies are hardly to monitor the status of market continually and intelligently. In order to reduce the total cost to get the analysis, many small to middle scale industries are now expecting a system which enables them to deal with dynamic information operations intelligently without incurring much cost.

Information filtering is related to processes such as routing (with a heritage in message processing), categorization, and extraction, as "current awareness" [5], as "data mining" [6], and as "Selective Dissemination of Information" [7]. Information filtering systems have been referred to as time saving devices [8]. The goal of information filtering is the prioritizing of information,

which assists the direction of human attention. Prioritizing can take the form of highlighting items of high importance or deleting items that are not considered relevant. Even though the term filtering has a literal connotation of leaving things out, we use it here in a more general sense that includes selecting things from a larger set of possibilities [9].

Information seeking is a very complex process [10] due to the extremely high variability of human behavior. As a result, process research is very important for this area [11].

While real time business intelligence has been demanded, the authors of this paper have proposed another information retrieval process with better performance and via an agent based framework [15].

VII. Conclusion and Discussion

In this paper, an agent-based security business data integration middleware for heterogeneous enterprise legacy sub-systems has been proposed. The framework of the middleware, the security consideration and the detail implementation of ECA agent and communication agent has been introduced.

The results from the case study showed that the research is of great help to the middle enterprises in saving their computing resource while achieving the information integration.

However, there is an assumption that the applications are using traditional database management system including Mysql, Sqlserver, Oracle, Sybase and DB2. Some other data storages on desktop such as excl, dbase, etc. are not tried yet and additional research will be performed to cover the other kinds of data storage.

Acknowledgment

This research is supported by Zhejiang Gongshang University, China. The authors want to express their special thanks to the team members; the research work will be exciting and interesting with their great effort and wonderful comments.

This research was financial supported by National Natural Science Foundation of China with No. 60873022 to Dr. Hua Hu, The Science and Technology Department of Zhejiang Province, China with No.2008C23034 to Dr. Hua Hu, No. 2008C11009 to Dr. Bin Xu, and No. 2008C13082 to Dr. Xiaojun Li, and Education Department of Zhejiang Province with No. 20061085 to Dr. Bin Xu.

References

[1] G. Salton, *Automatic Information Organization and Retrieval*, McGraw-Hill, New York, 1968.

[2] N.J. Belkin and W.B. Croft. Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35(12): 29-37,

[3] P. Denning. Electronic junk. *Communications of the ACM*, 23(3): 163-165, March 1982.

[4] D.W. Oard and J. Kim. Implicit feedback for recommender systems. In Recommender Systems, Papers from the 1998 Workshop, Technical Report WS-98-08, pages 81-83, *Madison, WI. Menlo Park, CA: AAAI Press*, 1998.

[5] P. Leggate. Computer-based current awareness services. *Journal of Documentation*, 31:93-115, 1975.

[6] D.W. Oard and G. Marchionini. A conceptual framework for text filtering. Technical Report EE-TR-96-25, CAR-TR-830, CLIS-TR-96-02, CS-TR-3643, *University of Maryland, College Park*, May 1996.

[7] T. Yan and H. Garcia-Molina. Distributed selective dissemination of information. In P*roceedings of the Third International Conference on Parallel and Distributed Information Systems*, pages 89-98, IEEE Computer Society, September 1994.

[8] P.E. Baclace. Information intake filtering. In *Proceedings of Bellcore Workshop on High-Performance Information Filtering*, Morristown, NJ, November 1991.

[9] T.W. Malone, K.R. Grant, F.A. Turbak, S.A. Brobst, and M.D. Cohen. Intelligent information sharing systems. Communications of the ACM, 30(5): 390-402, May 1987.

[10] Ellis, D. (1989). A behavioral approach to information retrieval system design. *Journal of Documentation*, 48(3), 171-212.

[11] Vakkari, P. (2003). Task-based information searching. *Annual review of information science and technology*, 37, 413-464.

[12] R. Kalakota, M. Robinson, *Dual-shore project management: Seven techniques for coordinating onshore-offshore projects*, http://www.informit.com/articles/article.asp?p=409917, 2005

[13] K.C. Jones, *Poor Communications, Unrealistic Scheduling Lead to IT Project Failure*, http://www.informationweek.com/news/showArticle.jhtml?articleID=198000251

[14] T. Finin, Y. Labrou, and J. Mayfield. KQML as an Agent Communication Language. In *Software Agents*, pages 291–316. MIT Press, 1997.

[15] H. Hu, B. Xu, "An Agent-based Framework for Intelligent and Dynamic Business Information Retrieval," *Workshop on Intelligent Information Technology Application (IITA 2007)*, 2007, pp. 77-81

Bin Xu (Tiantai, Zhejiang Providence, China, 1972) received Ph.D. degree on computer science from College of Computer Science & Technology, Zhejiang University, Hangzhou, China in March, 2005; received Bachelor degree on computer application from Department of Computer Science & Technology, Zhejiang University, Hangzhou, China in June, 1993.

He had been a software engineer from 1993 to 2005, and served as the software development manager in a famous software vendor for hotel management information system from 1995 to 1999. He led a software reengineering project on an online trading system. He is currently with Zhejiang Gongshang University, Hangzhou, China as an Associate Professor and vice director of the Department of Software Engineering. The work in this paper belongs to “Collaborative Commercial and Manufacturing” project supported by the Science and Technology Department of Zhejiang Province, China. His previous research interests include software project management, software engineering process improvement, and cost efficient protocol on sensor network. His current research interests include role based collaboration in software engineering and online collaborative learning.

Dr. Xu is a member of IEEE and belongs to the society of Control Systems. He has published around 30 academic papers which included in IEEE digital library.