Winter 12-5-2004

A Web-based Operation Management System for Distributed Divisional Organizations

Wei Shang

Yijun Li

Qiang Ye

Wenjun Sun

A Web-based Operation Management System for Distributed Divisional Organizations

Wei Shang, Yijun Li, Qiang Ye, Wenjun Sun

School of Management, Harbin Institute of Technology, Harbin 150001, China
{shangwei, liyijun, yeqiang, wjsun}@mail.hit.edu.cn

ABSTRACT

Operation Management is an important and complex task for a divisional structured organization, especially when the divisions are distributed geographically. In most cases, such organizations didn't not urge all of it's divisions to use an integrated information system at the very beginning. But with the development and the expanding of the organization, they sometimes found themselves in the trouble of information exchange and almost lost control of their divisions. At such time, however, on one hand the head quarter inquires more detailed information and more business control on the divisions. On the other hand, some divisions are well built and have its own business processes and information systems. It's impossible for them to rebuild the information system to integrate with the other divisions and the head quarter as well. Operation Management System (OPMGT) enables real-time inspection of the divisions' operational data and flexible operation evaluation of each division via the Internet and without much change on the other information systems. The OPMGT presented in this paper was originally developed for the head quarter of a distributed divisional based organization to govern the distributed divisions. System analysis, design and implementation of OPMGT are discussed in detail. Having been developed on the basis of eFramework, a J2EE framework, OPMGT is proved to be highly sufficient in operation management of a distributed divisional structured organization, and it may also do some help to integrate information systems in some degree.

Keywords: Operation Management System, Distributed divisions, Enterprise Information System, J2EE framework, Operation Evaluation

1. BACKGROUND

With the development of information technologies, Enterprise Information System (EIS for short) has been playing an important role in the enterprise management. The new generation of eBusiness (electronic business) becomes more rational and focuses more on the IT (Information technology) revolution inside the organizations [1]. And the enterprises tend to be geographically distributed with the development of the global economics. To control the operation of the whole organization, information system together with the Internet is a possible facility. There are three basic organization structures in practice: functional, divisional and matrix. Different kind of organization structure calls on different emphasis in operation management.

In a functional structure, each part of the organization should work together in a coordinated manner and operational management should be performed upon the vision of the whole organization. While in a divisional structured organization, each division is quite independent financially in most cases, and the top management usually does not interfere with the operation details of each division, but tends to supervise their operation situation as a whole. Matrix structured organization simultaneously implements both structures mentioned above by having a division manager and a functional manager at the same time to keep control of productive and managerial affairs.

The operation management function in most existing EIS is designed for functional structured organization. And in a matrix-structured organization, project management is the commonly used and effective method to manage the operation of the organization. As for the divisional organization, unlike the other two types, cooperation among divisions is loose and organizations today might have their divisions in different cities or even different countries. Unfortunately, in many geographically distributed organizations, operation management is usually emphasized inside each division only, and operation management of the whole organization is neglected. The importance of developing an effective operation management system for distributed divisional structured organizations has been recognized. We have built such a system, called OPMGT, which is web-based and tailored for a design company of the petroleum industry to mange their distributed divisions.

System requirements are discussed in the following section. The eFramework on which we built our OPMGT is introduced in the third section. Then there goes the detailed design and development. Implementation and conclusions are discussed as the last part of the paper.

2. SYSTEM REQUIREMENTS

OPMGT is an information system for operation management, especially for distributed divisional structured organizations. Here, we limit our research in

the organizations that have more than three divisions and tend to control the operation of each division at a relatively high level instead of a detailed level. Since the divisions might be distributed geographically, we should build our system as an Internet based application. Only in this way, different divisions can access the system easily and interact with the head quarters continently. To understand what problems the OPMGT solves and how the OPMGT works, we need to take a look into the operation system requirements and infrastructure.

To supervise the divisions and make a future strategy, the top executives of a divisional organization usually care about the following aspects of the divisions' operational affairs.

- How do they fulfill their part of the whole organization's goal? Do they make more profits or do they spend more over the annual budget? Finding the weak points of the organization's operation is essential to improve the performance of the whole organization.
- How to evaluate their operation effects compared with other divisions? Bonus distribution among the divisions is not an easy task. You cannot just scatter the money according to the profit per person. Some divisions do not make much profit or even do not make any profit at all, but their contributions to the overall organization are the same or even more.
- How to make the anticipated profit or cost-based budget of the next year referenced with their historical operation data? Identifying the increase or decrease in efficiency is due to accident or an unavoidable trend is a rather statistical problem that needs the support of a large amount of operational data.
- How the division manager's leadership affects the whole division's operating situation? Since the division is independent in many aspects, the divisional manager is quite important to the performance of the whole division.

The employees in each division are also concerned with their performance in some of or all of the above aspects. The key of operation management in divisional structured organizations is how to evaluate each division. This measurement should be fair and open.

Technically, we adopt a Web-based J2EE framework to build the OPMGT as a browser and server (B/S) application. In this way, the system can be accessed anywhere through the Internet and has the enterprise features of expansibility and reusability.

3. A WEB-BASED J2EE FRAMEWORK

3.1 Infrastructure

The JAVA 2 Enterprise Edition (J2EE) is one of the most popular technologies used in EIS development by its features of platform-independent, object-oriented, component-based and multi-tier-supported. There are more and more enterprise applications developed totally or partly using the J2EE technology. J2EE framework is a set of tools to enhance the efficiency and reliability of the enterprise application development. Here we introduce a web-based J2EE framework, eFramework (developed and maintained by Polysoft International Corp.), which is especially designed for EIS.

J2EE introduced a component-based method to design, develop and deploy enterprise applications. Despite of the various J2EE design patterns, J2EE infrastructure is composed of three tiers: client, server and background system (e.g. the database system). The J2EE specification defines the following component:

- Client component, which is a web page browser to parse the HTML document and support the JavaScript dynamic effects.
- Server component, which contains Servlets and JSPs.
- Business Component, which is possibly implemented by EJBs.

eFramework supports the new stable version of JSP and includes the newest Servlet APIs. EJB is included by eFramework as a solution for complex business logic. Applications based on eFramework can be deployed to any application server that supports the J2EE specifications.

3.2 The MVC design pattern in eFramework

The Model-View-Controller (MVC) design pattern has been recognized as a brilliant software design pattern that enforces the separation between the input, processing, and output of an application. eFramework adopts MVC pattern in web-based applications with J2EE technology. (Shown as Figure1.)

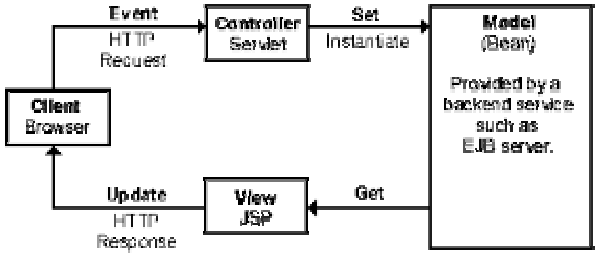

Figure1. M-V-C design pattern of eFramework

When a HTTP request reaches the web server, the following events occur:

- All the requests are captured by the Controller.
- Controller receives the request from a browser and decides where to send it by parameters assigned before hand.
- Controller sends instructions to Handler Factory and demands for a specific Handler to process the request.
- Controller commands the Handler to deal with the request.

- Handler presents the Model of M-V-C pattern, traces the states and implements the business logic. Each Handler knows its duty of combining multiple data beans into a Page Bean.
- Handler asks Persistence Manager for data and put these data into a web Page Bean.
- Persistence Manager is responsible for saving and getting data through JDBC and hides the detail of stateless session beans to implement the encapsulation rules of software engineering.
- If any exceptions occur during the process above, the Handler will decide whether to generate exception pages, for instance, error page beans specially designed for present error messages.
- Handler passes the assembled Page Bean to Controller.
- The Page Bean, which has been sent to the Controller, will give out its Presenter.
- The Controller passes the user request and response parameters to the Presenter, and asks them to present the Page Bean.
- The JSP page appointed in the Presenter only needs to get data from the Page Bean, convert them into HTML document and send it back to the user.

The whole closed loop is shown in Figure 2:

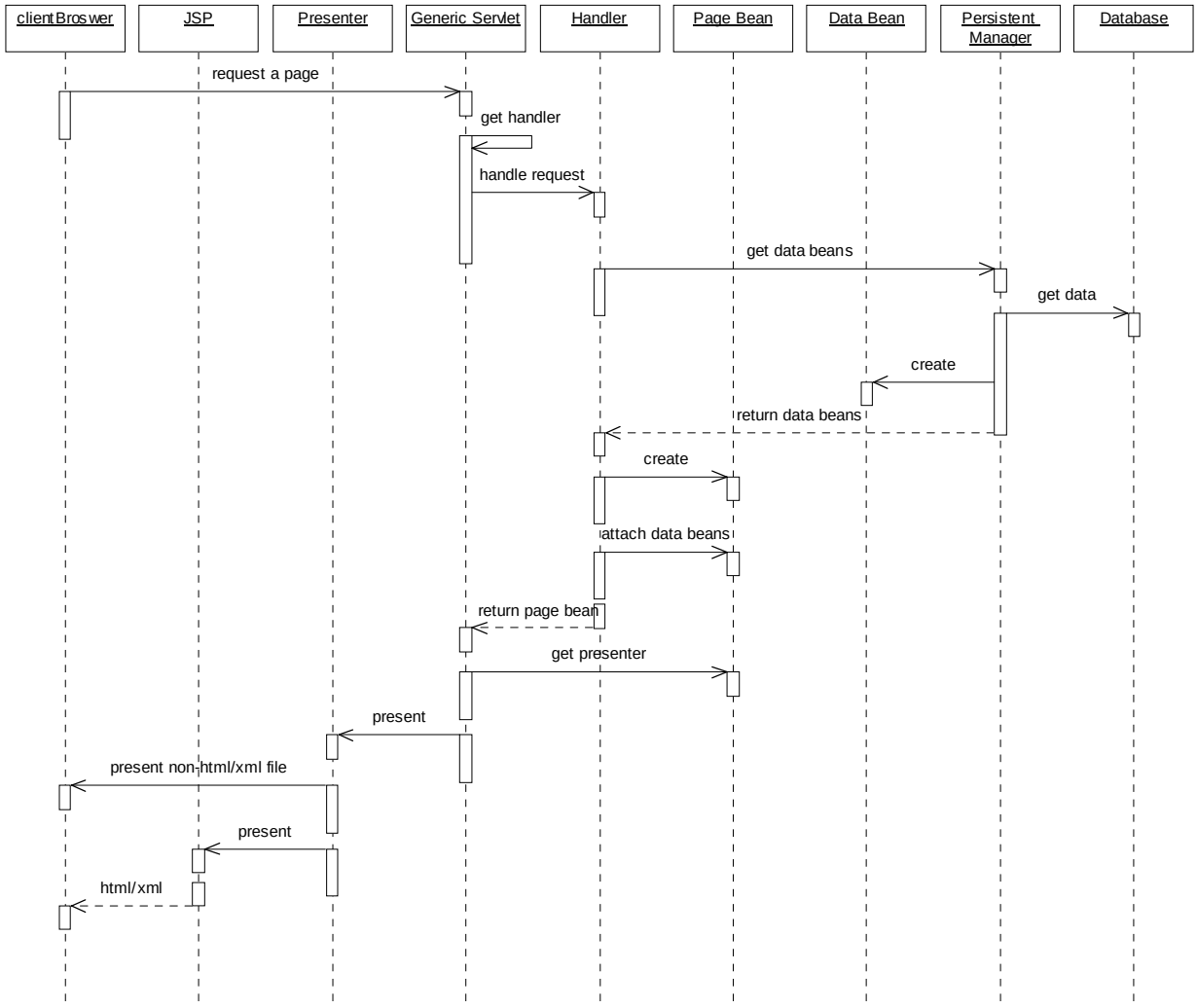

Figure2. M-V-C design pattern of eFramework

3. 3 Features

Presentation, business logic and persistent data management are separated clearly in the eFramework. Multi-language, customizing, and application diagnostics are supported as well. By supporting the mechanism of independent development on JSPs and Servlets, eFramework also enables non-Java developer to built JSP pages. Multi-tiered logging functions enhance the just-in-time inspection and effective debugging on the system. J2EE infrastructure enables flexible deployment on any Servlet container and JSP Engine compliant with the J2EE specifications. Although eFramework seems more complicated than other J2EE frameworks, such as struts and cocoon etc., it is more suitable for building large EIS for its outstanding features.

4. SYSTEM DESIGN

4. 1 System structure

Main function of OPMGT is to help the executives take control of the operation management. OPMGT should be a subset of an EIS, therefore coordination with other part of the EIS must also be carefully considered. The system structure is shown in Figure 3.

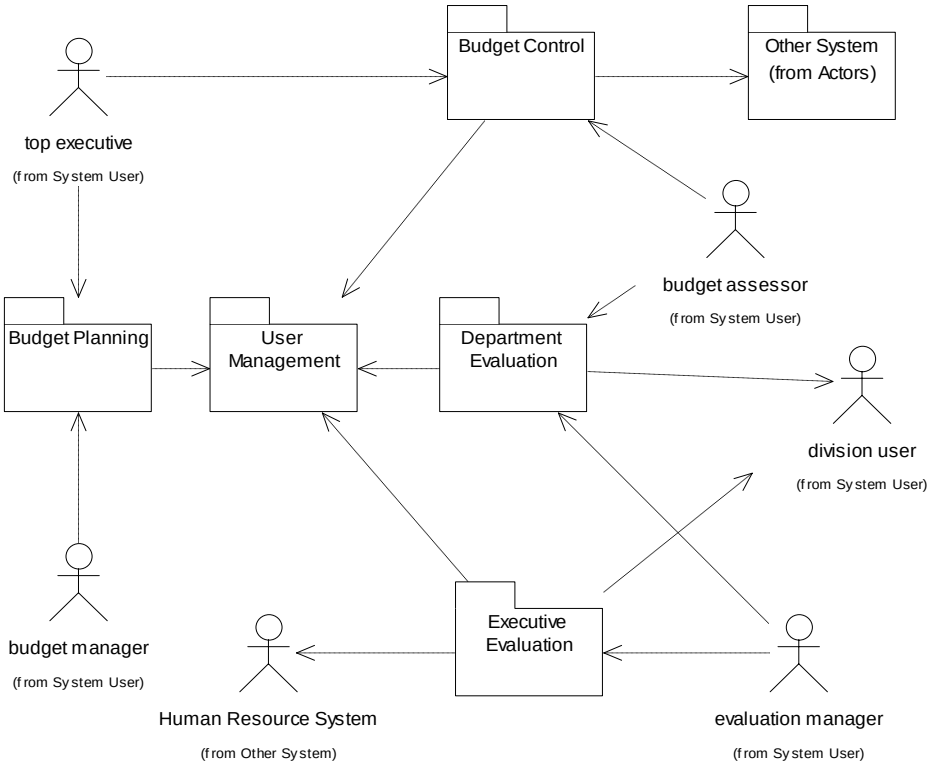

Figure 3. System Structure Diagram

Five packages are needed to satisfy the system requirements: user management, budget planning, budget control, department evaluation, and executive evaluation.

4. 2 Database management strategy

eFramework provides a database connection pooling management mechanism to manage the JDBC database connections. Since OPMGT is related to several databases, several connection pools should be built for the system to contact with the background databases. Information of these connections is specified in a special type of configuration files (java property files to be specific). This kind of file may have its content changed without having to force rebuilding of the whole project. Therefore, agile development on database connection management is implemented.

Small data unit, such as user information, is designed as a Java class. A persistent manager is a final class to manage the instances of those classes that interact with the databases to obtain small data units. This helps to encapsulate the database access by letting eFramework manage the lower level details of data exchange and ensure data transfer security and data consistency.

In OPMGT, historical data may be required for querying. We design a backup database to store a more recent historical data, for example, a two-year old dataset. When a request for historical data occurs, a connection to the backup database is built dynamically. After the processing of historical data, the connection is released automatically, since there is not a single active communication for a certain time. With the aid of eFramework, developers may concentrate on the database management logic and leave the routine database communication to eFramework.

4.3 User authentication and authorization design

User authentication and authorization is pretty indispensable in OPMGT. Without authentication, a user can only access very limit information, for most of the operation data of an organization could be confidential. Authentication is not enough, in the case that users of

different managerial levels need to have different levels of rights to use the system. A top executive checks raw operating data of each division but cannot modify it. While a common division user may evaluate the operation of his division but cannot change the evaluating method. Authorization is complex and may change often during the run-time of the system also.

We therefore design two inheritable handlers: AuthenticatedRequestHandler and AuthorizedRequestHandler. The former class contains the default user authentication logic that ensures that the user has been logged into the system with right user ID and password. The latter class checks if a user has the right to do a certain operation by checking whether the request identity sent by the user has been assigned legal for the user to execute. The relations between user IDs and corresponding request identities are stored in a table. Only the system administrator has the right to maintain this information.

By inheriting from the AuthenticatedRequestHandler or the AuthorizedRequestHandler, each handler has its own security level. The build-in rules of authentication and authorization release the developer of each module from the trifle work of deciding the user's rights in every operation.

4.4 Detailed design of the system

4.4.1 Budget planning module

The budget-planning module is concerned with the operation control in the organization level. Making the annual income budget and outcome budget is the main function of this module. Tracing the implementation of the budget and compare the actual budget with the planning budget is another function of this module. An annual budget may contain several sub budgets for each division. Both the top executives and the division managers have the rights to check these budgets.

Making and tracing these budgets with a visual statistic chart may help all level of the managers to have an impressive understanding of the overall situation of the organization. We design a statistic figure generation package based on the JFreeChart library. A chart factory manages the instance of three kinds of charts: pie chart, bar chart and line chart. When a user requests to draw a figure, the chart factory offers a specific chart drawing class and return the user a JPG format figure. This statistic figure generation package can be used in other part of the OPMGT also.

4.4.2 Operation data management

There are two means to gather the operation data: collecting from the existing information system and inputting it into the OPMGT manually. If there's already electronic data in the EIS, this kind of data could be read from the enterprise database directly. But in some cases the data may be collected by commercial software that does not open its database. For example, the expense flow may have been collected by an existing accounting information system. In this instance, the OPMGT has to import some kind of output document from that software. The output document could be a Microsoft Excel file or a XML file or some other formatted data file. We design an XML import module based on Xalan APIs and an Excel import module based on JXL library to import existing operation data from existing EIS. If not all the operation data has been collected electronically, it's the OPMGT's responsibility to propose a convenient, user-friendly operation data input interface. An input form with combo boxes and check boxes is designed as a manually data input interface in OPMGT.

Querying and summarizing the operation data is also designed as an important function in operation data management module. Once the data has been input into the system, nobody can revise them or delete them. If there's any mistake in the raw input data, the only thing to correct it is to add a negative income or outcome noted the corresponding income or outcome ID. This rule just presents the common sense of accounting information system.

4.4.3 Division evaluation and executive evaluation

Evaluation measurement of the division's performance could be complicated and inconstant. Accordingly, we develop an equation parsing and calculating package. The equation is converted into a reverse Polish notation and a binary tree is used to present the equation and the sub equations. With the flexible equation mechanism the evaluation measurement can be defined and revised in the run time. Some variables in the evaluation equation are from the gathering of the operating data and others may be coefficients which value is different according to different year and different division. These two kind of variable are distinct in that gathering of the operating data is dynamic and could be obtained from the system automatically, while the value of the coefficients are decided subjectively by the evaluating operator. Thus we present and store these two kinds of variable separately.

The executive evaluation differs from the division evaluation only in the presentation of the equations and the source of the variable values. Therefore, we use the same module but different storage to design the executive evaluation module.

5. SYSTEM IMPLEMENTATION AND CONCLUSIONS

The OPMGT is developed among a small group consists five developers. We built a CVS depository to store the code and coordinate the team development process.

Taking the advantages of the eFramework, the development process is efficient and efficient. Testing and debugging work is also smooth according to the well-structured design pattern and the eFramework logging mechanisms. Since the OPMGT do not have heavy traffic, an Apache server with a tomcat plug-in is enough for a web application server.

In developing the OPMGT, eFramework is proved to be an efficient in information system development. But due to its powerful infrastructure, it is not easy for a developer to master the spirit and exert the best performance of the framework. The starting of our development is much harder than the following period. Consequently, a long term and relatively steady develop team may be better for building an information system using this eFramework.

The authentication and authorization mechanism is sufficient for the access control of the system functions. The flexible design of the evaluation equation enables various dynamic measurements of the divisions in an organization. The organization testing the OPMGT has found it practical and effectual in managing the operation of the divisions as a whole. Further more, as an application based on J2EE technology, the OPMGT has the cross-platform feature and this makes it possible to be integrated into existing enterprise system. The excel file input and figure output functions also make it convent to exchange data and express information in a distributed divisional structured organization. The design company who raised the request of building this system is very satisfied with it. However, they have not used it for long time. And with time past, more improvement of the system must be done.

ACKNOWLEDGEMENT

This research is sponsored by the Natural Science Foundation of Heilongjiang, PRC (G0304) and the Key Sci-tech Project Plan of Harbin, PRC 2004AC9CT073).

REFERENCES

[1] Erik Brynjolfsson, Glen L. Urban, *Strategies for e-business success*, MIT Sloan Management Review, Nov. 2001, pp, 38-43

[2] James I. Cash, Building the *Information-Age Organizations: Structure, Control, and Information Technologies*, McGraw-Hill Book Co. 1998, pp, 28-32

[3] Deepak Alur, Jhon Crupi, Dan Malks. "Sun Java Center J2EE Patterns", http://developer.java.sun.com/developer/technicalArticles/J2EE/patterns/, 2002.5.14

[4] Yan Hong, *Java and Patterns*, Publishing House of Electronics Industry, Beijing,