2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)
IPCSIT vol. 53 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V53.No.1.57

The Design and Implementation of Dynamic Customized Platform Based on Database Storage in ERP Secondary Development

Jiang Guihua[+] and Zhang Yu

Department of Computer Science and Technology
South China University of Technology
Guangdong, China

Abstract. We often encountered the conflict between the packages in the ERP secondary development and the upgrade patches from ERP provider in the ERP development process .In this paper, a dynamic customized platform basing on database storage is presented . Furthermore, the design and implementation in detail are expatiated. Our solution is to make the secondary development of functional modules independent of the original ERP system. We do not change the core code, but the data is derived from the ERP and saved into database by decorator pattern. This design leads to rapid development, eliminates the conflicts and enhances the scalability of system, to provide people a different solution for deploying ERP system successfully.

Keywords: ERP secondary developments, dynamic customized platform, upgrade patch, meta-data, decorator pattern

1. Introduction

Simply say ERP is the Enterprise Resource Planning, as early as the late 1990s.Nearly 80% of the world's top 500 enterprises using the ERP management software [1]. Mature ERP software contains the international advanced management ideas, but they cannot meet the special needs for domestic enterprises, so it should hand ERP software for secondary development and customization work [2]. Some mature ERP products have several years' history, and a large number of engineers were maintaining the software, as well as a large number of clients were using it. Those clients might find some bugs while using, and those bugs would be deleted in the standard system upgrade and patch package .On the one hand, the clients requested individual needs, which we called secondary development, on the other hand they would use upgrade or loading patch from ERP provider to enjoy these after-sale service [3]. So there always had two or more development teams who maintain and upgrade the same ERP product.

In the traditional development pattern, though modifying the metadata and jar package deployed on the server we can do secondary development and patch package to develop, maintain and upgrade the function of ERP [3]. However, we often modify the same business components while doing secondary development and patching up the bugs, it leads to conflicts between secondary development and patching [4].

This paper analyzes the conflicts between secondary development and patches in the implantation of ERP system,and gives a solution to get rid of the conflict . It's a dynamic customized platform model based on database storage management. We will store the secondary development into the database,and then we use static logic accessing and load the metadata in the database to realize the customized development .With

[+] Corresponding author.
E-mail address: jianggh2008@163.com.

this,we can completely get rid of the conflict and loose coupling between secondary development and patching and do more efficient development.

2. The Conflict between Secondary Development and Patching

After getting ERP software, the customers will make personalized needs, so the developers need to do ERP secondary development [5]. After ERP product is on-line, ERP providers will provide patch packages with upgrade, extension, maintenance and other services. ERP customers enjoy these services through patching packages [6]. As shown in Figure 1, the metadata, patching jar package is deployed on the server, the secondary development and customization is usually in the customer site.

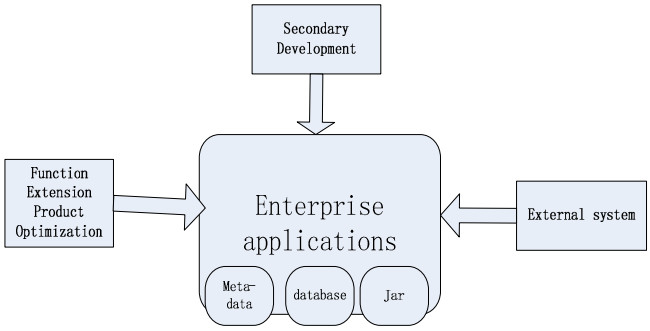

Fig1 ERP System Operating Framework

We can conclude the conflicts between secondary development and patching as following:

(1) the merging of meta-data and code difficult and inefficient[7].

(2) the testing engineers need to do regression testing again and again after merging and patching.

(3) it's difficult to quickly locate the sources whether the bugs is from the secondary development or patching.

(4)it is very difficult to do unit testing.

3. The Dynamic Customized Platform

The process of the secondary development in tradition is like this: the secondary development package is loaded firstly, the patching package code is achieved by merging the meta-data and code manually or automatically. There are many disadvantages in this second development, such as: the cycle of the development and deployment is too long, the process of merging often makes problems, and the outcome of the secondary development is reused difficultly. This pattern of development is too invasive and costs high [8].

We can define the dynamic customized platform as this: The End User and IT specialists in the customers and the developer team don't need to rebuild the application of ERP system application or restart the server .They can adjust structure and behavior in application to achieve the customization, they can develop many characteristics packages based on the core application, and these packages can rely on each other and can be used together which lead to personalized customization. The main idea is to increase and adjust the meta-data to customize the existing business process, in which resulting Java scripts or code or configuration files are stored in the database. If the java scripts cannot realize some complex business process, we can add some class to decorate the original business components, so we can meet customers' specific needs. As the new classes is not present in the original jar package, so they loosely couples with the patching packages completely.

4. The Specific Implementations

We can expand meta-data and business logic to realize the dynamic customized platform [8].

4.1 The Meta-data Expansion s

As showed in Figure 2, there will generate some metadata fragments while modifying the relevant metadata or adding relevant meta-data on the platform, we can use ORM tools to interact with database by pojo (Plain Old Java Object, the specific data structure), and then with the meta-data engine we can merge these fragments stored in the database as files into a whole piece of data, so we complete the process of meta-data personalized expansion and meet customers' individual requirements[9]. With this the adjusted meta-data is stored in database instead of as files .We don't need to modify the document meta-data. In this way, we can use hot-deploying and dynamically loading the customized meta-data fragment to do some modifications on the metadata while the server is running, then we save the time of restarting the server, especially we can do unit testing with a ease[10]. Any modifying about the products can be done on the framework, so it provides high quality and efficiency development model with us.

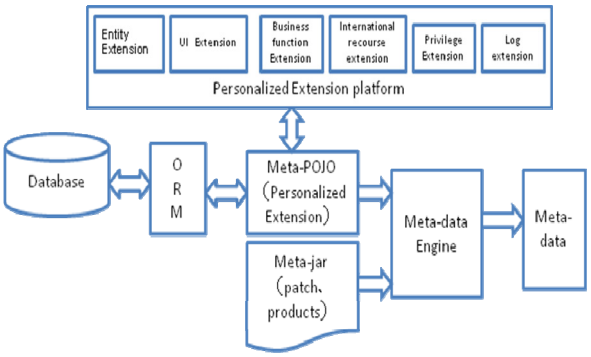

Fig 2 The Meat-data Extension Framework

4.2 Business Logic Extension

The code extension includes the java classes and java scripts. We use another class which has achieved a specific interface to realize the extension of business logic and record the combination relationship by configuration files between the two, in which we can dynamically modify the metadata in the system and edit the java scripts, and store the adjustment information as POJO to the database

a) Using Java Scripts

We can use java scripts to achieve some simple business components, such as adding some components to the UI [11]. The scripts are loaded with a string instead of generating a class by the script engine. These strings can be saved to the corresponding meta-data nodes. We store configuration information and metadata (including java script) of the code plug-ins in the secondary development into the database, and then use a combination to package the corresponding business logic components and business template

b) Using Java classes

In addition to meta-data extension in the customized development such as adding entity field[12][13], UI controls, controls binding with the entity properties and view display extension, there are some complex business logic extensions that need to use Java classes, we realize this by adding the related personalized methods around the business components[14][15]. In tradition, we always modified or extended the compiled class in the secondary development, it's has risks on that we could not sure about anti-compile class was as the same as the original source. In the dynamic customized platform we use combination to replace the origin, which can meet our customers' needs [17][18].

4.3 Customized Development

There are two ways that we realize the customized development, the one is to modify the product based on in the original product. The other is redefinition

a) General Business Templates

The model in Figure 3 is fit for the above two methods. The template in the following picture is an abstract of the common business operations, we extract and code plug-in mechanism and metadata extension

mechanism for interaction, leaving the interface to delegate code plug-ins and scripting engine implementation. By this, we just need to define business templates to complete a number of business units [25] [26].

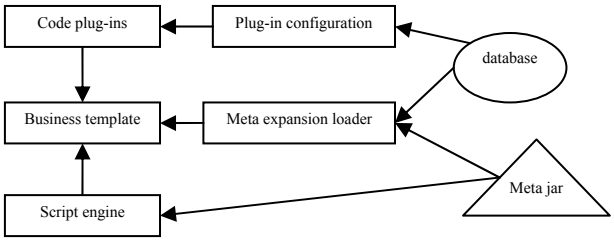

Fig 3 business expansion framework

b) Extending the Business Template by Decorator Pattern

If we modify something based on the original product, we must modify the base classes on the application framework, we can make the base class into a business module and use Decorator pattern, which need the support of the meta-data nodes extension and the code plug-ins and scripts engine. The only differences in different clients using the same ERP products are configuration files in the server and the data stored in the database. With this, we can do unified planning to facilitate ERP provider and will not integrate the customization into ERP system, which leading to reducing stress about ERP system[30][31].

Taking UI for an example, as showed in Figure 4, the UI template TemplateView achieves the universal business logic. Extend meta read the corresponding meta-data fragments from the database according to the context and re-adjust the UI and register the corresponding events for the controls according to the metadata fragments. When the user triggers event, TemplateView will assign the js Engine to deal with js. Plug-In decorator read the corresponding plug-in configuration and call the business logic by reflection. With the model. We can save secondary development into the database entirely [35].

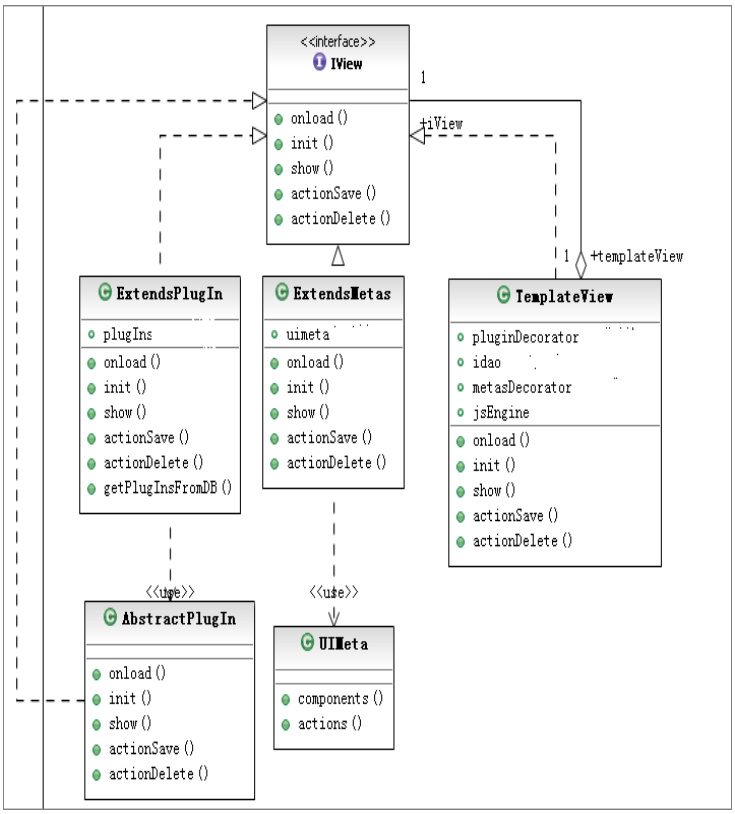

Fig 4 business template component class

5. Summary

We get rid of the conflicts between the secondary development and patch by saving the extension info of the secondary development into a database, and we can import or export any the secondary development by sql strings, which achieve that enabling or disabling all the running secondary development. With this loosely coupled development mode, we reduce the workload of merging the metadata and code, we can quickly locate the problem whether from the patch or from the second-ary development and eliminate the risks of conflict be-tween the secondary development and the patch. The system can be hot-deployed, and the do unit testing with a ease. All of these can make an efficiency development. We believe that it is very widely used in the development of ERP systems.

6. References

[1] Bjorn Johansson, Mike Newman. Enterprise Information Systems: Competitive advantage in the ERP system's value-chain and its influence on future development. Feb. 2010.

[2] Dasha Hu, Zhishu Li,Yunzhu Ni. IEEE Conferences Integration of Heterogeneous System Based on Business Object Management and Service Science, 2009. MASS '09. International Conference on.

[3] Ross, J. Dow Coming Corporation: Business Processes and Information Technology. J. Info. Technology 15, 3, (Sept. 1999).

[4] Rittammanart, N. Wongyued, W. Dailey. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology:ERP application development frameworks: Case study and evaluation.Aug.2008.

[5] Ming Xue,Changjun Zhu,Shilong Song. Control, Automation and Systems Engineering:The Development and Implementation of Household Textile Industry ERP System Based on B/S. 2009-07.

[6] Liang-Chuan Wu,Yao-Wen Hsu,Chorng-Shyong Ong.International Journal of Information Technology and Management: ERP implementation: a quantitative model for organisational learning. Apr, 2007.

[7] Abdinnour-Helm, S., Lengnick-Hall, M.L. and Lengnick-Hall, C.A. (2003) 'Pre-implementation attitudes and organizational readiness for implementing an Enterprise Resource Planning system', Journal of Operational Research, Vol. 146, pp. 258-273.

[8] Gattiker, T.F. and Goodhue, D.L. (2004) 'Understanding the local-level costs and benefits of ERP through organizational information processing theory', Information and Management, Vol. 41, pp. 431-443.

[9] Wang, Y.C., Heng, M. and Ho, C.T. (2005) 'B2B integration as the outcome of business network determinants in a relational model', International Journal of Production Planning and Control', Vol. 16, No. 6, pp. 575-585.

[10] Botta-Genoulaz, V., Millet, P. A., and Grabot, B., A survey on the recent research literature on ERP systems. Computers in Industry, 56, 6 (2005), 510--522.

[11] Daneva, M. and Wieringa, R., Cost estimation for cross-organizational ERP projects: research perspectives. Software Quality Journal, (2008).

[12] Monnerat, R. M., de Carvalho, R. A., and de Campos, R., Enterprise systems modeling: the ERP5 development process, in Proceedings of the 2008 ACM symposium on Applied computing. 2008, ACM: Fortaleza, Ceara, Brazil.

[13] Olinger, C., The Issues behind ERP Acceptance and Implementation, APICS: The Performance Advantage,

[14] R. E. Johnson, “Components, frameworks, patterns,” in SSR’97: Proceedings of the 1997 symposium on Software reusability. New York, NY, USA: ACM, 1997, pp. 10–17.

[15] M. E. Fayad, D. S. Hamu, and D. Brugali, “Enterprise frameworks characteristics, criteria, and challenges,” Commun. ACM, vol. 43, no. 10,pp. 39–46, 2000.

[16] A. Gerdessen, “Framework comparison method: Comparing two frameworks based on techincal domains focussing on customisabilityand modifiability,” Master’s thesis, University of Amsterdam, August 2007.

[17] M. J. Yuan and T. Heute, Seam: Next-gen Web Framework. Prentice Hall, 2007.

[18] Apache Software Foundation, “OFBiz, the Apache Open for Business Project — Open Source E-Business / E-Commerce, ERP, CRM, POS,” 2008. [Online]. Available: http://apache.ofbiz.org

[19] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.

[20] M. Fowler, Patterns of Enterprise Application Architecture. Addison-Wesley, 2002.

[21] J. Verville, A. Halingten, An investigation of the decision process for selecting an ERP software: the case of ESC, Management Decision 40 (3) (2003) 206 – 216.

[22] E.T.G. Wang, T.-C. Ying, J.J. Jiang, G. Klein, Group cohesion in organizationalinnovation: an empirical examination of ERP implementation, Information and Software Technology 48 (4) (2006) 235 – 244.

[23] J. Ward, C. Hemingway, E. Daniel, A framework for addressing the organizational issues of enterprise systems implementation, Journal of Strategy Information Systems 14 (2) (2005) 97 – 119.

[24] J.W. Weiss, D. Anderson, Aligning technology and business strategy: issues and frameworks, a field study of 15 companies, in: Proceedings of the 37th

[25] Hawaii International Conference on System Sciences (HICSS 2004), January 2004.

[26] N. Welti, Successful SAP R/3 Implementation: Practical Management of ERP Projects, Addison-Wesley, Reading, MA, 1999.

[27] L.-C. Wu, C.-S. Ong, Y.-W. Hsu, Active ERP implementation: a real options perspective, Journal of Systems and Software 81 (6) (2008) 1039 – 1050.

[28] R.K. Yin, Applications of Case Study Research, second ed., Sage Publications,Thousand Oaks, CA, 2003.

[29] R.K. Yin, Case Study Research: Design and Methods, fourth ed., Sage Publications, Thousand Oaks, CA, 2009.

[30] S. Abdinnour-Helm, M.L. Lengnick-Hall, C.A. Lengnick-Hall, Preimplementation attitudes and organizational readiness for implementing an Enterprise Resource Planning system, European Journal of Operational Research 146 (2) (2003) 258 – 273.

[31] Aberdeen Group, Vertical Industry Solutions: Baan Leads in Innovation, Market Viewpoint 10, July 1, 1997. <http://www.aberdeen.com>.

[32] G.B. Alleman, Agile project management methods for ERP: how to apply agile processes to complex COTS projects and live to tell about it, in: Extreme Programming and Agile Methods: XP/Agile Universe 2002, Lecture Notes in Computer Science, vol. 2418, 2002, pp. 70 – 88.

[33] M. Backman, C. Butler, Big in Asia: 30 Strategies for Business Success, Palgrave Macmillan, Hampshire, UK,

[34] M.S. Jayaraman, Implementing BPR and ERP, Business Line, May 3, 1998.

[35] <http://www.angelfire.com/ct/etmethodology/erpnews4.html>