

# Development of Enterprise Information System Using Business Component-Based Approach

Xueping Wang<sup>1</sup>, Chenghu Zhang<sup>1</sup>, Xinqin Gao<sup>2</sup>, and Zongbin Li<sup>2</sup>

1 School of Economics and Finance, Xi'an Jiaotong University, Xi'an 710061, China

wangxueping\_2008@yahoo.com.cn, zch@mail.xjtu.edu.cn,

WWW home page: <http://www.xjtu.edu.cn/en/academics&research/ef.htm>

2 School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, China

gaoxinqin@yahoo.com.cn, lzb9910@mail.xjtu.edu.cn,

WWW home page: <http://www.xjtu.edu.cn/en/academics&research/me.htm>

**Abstract.** Facing the great challenges from various changes, the traditional enterprise information system is becoming obsolescent. The timely response to market changes has become the competitive advantage. In this paper, business component-based development of modern enterprise information system is studied. Based on the business component and its technology, business component-based software development process is proposed. A clustering algorithm is used to identify the business components from requirement model and a rules library is used to assemble the business components to a business application. Component-based hierarchy architecture of enterprise information system is established, which are software infrastructure layer, middleware container layer, business entity layer, business process layer and application interface layer. The business component-based enterprise information system has the rapid reconfigurability, reusability and extensibility to adapt itself to the changes of business environment.

## 1 Introduction

In the 21<sup>st</sup> century, enterprise confronts the great challenges from unprecedented and abrupt changes, including economic globalization, saturated market and rapid IT progress. Under the pressure of market competition, the timely response to market changes becomes the competitive advantage [1]. The enterprise information system

---

*Please use the following format when citing this chapter:*

Wang, X., Zhang, C., Gao, X., Li, Z., 2006, in International Federation for Information Processing, Volume 205, Research and Practical Issues of Enterprise Information Systems, eds. Tjoa, A.M., Xu, L., Chaudhry, S., (Boston:Springer), pp.361-369.

must have the characters of rapid reconfigurability, reusability and extensibility to adapt itself to the changes of business environment.

The object-oriented technology is primarily adopted in the development of enterprise information system [2, 3]. However, the enterprise information system based on O-O technology needs to open the source code of reusable object to system developer, which belongs to the white-box mechanism [4]. In order to reach the black-box reuse, business component-based approach is proposed and currently used in enterprise information system [4, 5]. Such technology hides the implementation process of function, and only provides interface publicly to implement the development and reconfiguration of system. With the integration, dynamic and reconfigurability, business component-based development approach has been the popular approach to software development [6].

The aim of this paper is to develop the enterprise information system using business component-based approach. Section 2 introduces the concept and characteristic of business component technology. Section 3 describes the business component-based development approach and its two main stages: business component identification and business component assembly. Section 4 uses business component-based approach to develop enterprise information system and the component-based hierarchy architecture of enterprise information system is established. Section 5 summarizes this paper and draws conclusions.

## **2 Business Component and Business Component Technology**

### **2.1 Business Component**

A software component is defined as the independently developed and deployed software unit having well-defined interfaces to perform specific functions [7]. A business component represents a software implementation of an autonomous business concept or business process. It is composed of all software components necessary to express, implement, and deploy a business component as an autonomous, reusable element of an information system [8].

The business components are the autonomous, independent and large granularity software units and easy to assemble into business application to implement the real world business concepts [7]. Furthermore, the business components can be reused in the same or similar domains. They export functionality to their environment and may also import functionality from their environment through well-defined interfaces.

Business components vary from traditional reusable software artifacts, such as code segments and objects [9]. In the granularity level, the traditional reusable artifacts are the low-level technical-oriented representation of the domain, while components are intended to provide a high-level business-oriented representation of the domain. Moreover, a component is a self-contained executable program to provide a specific service and has an interface to communicate with other components.

## **2.2 Business Component Technology**

Business component technology is the new progress of component technology. It has four primary characteristics [10]:

(1) Advancing the industrialization of software production

Business component technology provides the theory and the technology for the industrialization of software production.

(2) Saving the time and cost of software development

Using business component technology, the software development stages are completed in parallel way with high efficiency.

(3) Minimizing the gap between the business and the technology

The business component is a combination of business object and component. It is convenient for the smooth transition from business architecture to technology architecture.

(4) Supporting the rapid reconfiguration of information system

Business components are assembled into the full-scale business application. By means of plug & play, business components can be reconfigured as the business changes happen.

## **3 Business Component-Based Development Approach**

### **3.1 Component-Based Software Development Process**

Component-based software development (CBSD) is an emerging paradigm of software development [6]. Component-based development approach is thought to be the latest outcome to tackle software crisis [11]. It advocates an approach that the designer can identify the components that satisfy their business requirements from business components and assemble them into full-scale business applications [6].

As shown in Fig. 1, component-based software development process consists of two main stages: component fabrication (building the business components) and application assembly (building a business application from components) [9].

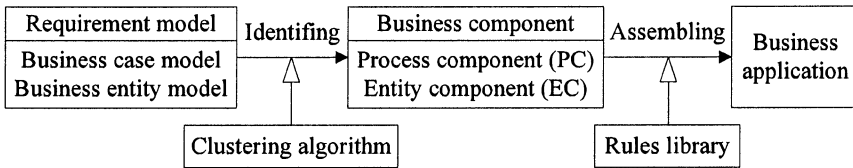


Fig. 1. Component-based software development process

### 3.2 Business Component Identification Using Clustering Algorithm

The component fabrication stage of CBSD consists of various phases. Business component identification is recognized as one of the greatest important phase of component fabrication stage [9]. Here, a clustering algorithm is used for identifying the components from a requirement model, which has been done using an object-oriented approach UML.

The unified modeling language (UML) is a visual and graphical modeling language. It can analyze and design the object-oriented system. The requirement model based on UML consists of business case model and business entity model. The core contents are business cases and business entities [11]. The goal of business component identification is to determine the high cohesive and low coupling business components from business cases and business entities of requirement model.

The clustering algorithm is a new multivariate statistical analysis method. It groups the data objects into the different clusters. The data objects of the same cluster have high similarity while the data objects of the different cluster have distinct difference. The cluster algorithm is described as follows [11, 12]:

(1) Denoting all the samples as a set,  $X = \{x_i \mid i = 1, 2, \dots, n\}$ . Where  $n$  is the number of samples.

(2) Determining the computational methods of  $R_{ij}$ , which denotes the relationship strength of the samples  $x_i$  and  $x_j$ .

(3) Computing the value of samples' relationship  $R_{ij}$  and establishing the relationship matrix  $D$ .

(4) Selecting appropriately a minimal value of samples' relationship  $R_{\min}$  as a criterion to execute the clustering process. If  $R_{ij} \geq R_{\min}$ , group the samples  $x_i$  and  $x_j$  into the same cluster. If the samples  $x_i$  and  $x_j$  belong to the same cluster and the samples  $x_i$  and  $x_k$  belong to the same cluster, group the samples  $x_i$ ,  $x_j$  and  $x_k$  into the same cluster.

The requirement model based on UML is taken as the data source of clustering algorithm samples, and all the samples grouped into the same cluster can be transform a business component.

UML use case diagram and activity diagram are used to describe the business case model. The business components identified from business case model are called the process components (PC), which are the encapsulation of business process and business rule [11]. The business cases are recognized as the samples set. Three

relationships may exist between two business cases. These are include, extend and generalization. If there is no relationship between the business cases  $x_i$  and  $x_j$ ,  $R_{ij}=0$ .

UML class diagrams are used to describe the business entity model. The business components identified from business entity model are called the entity components (EC), which are the encapsulation of business entity and business rule [11]. For computing the value of relationships between classes, static and dynamic relationships are used. Static relationships are computed based on the associations between classes and the dynamic relationships are computed based on use cases diagram.

The value of the static relationship  $RS_{ij}$  between classes  $x_i$  and  $x_j$  be defined as follows [9, 11, and 13]:

$$RS_{ij} = \sum_{y \in Y} (U_{yi} \cdot U_{yj} \cdot W_y) \quad (1)$$

Where  $Y$  is the relationship set of business entities.  $U_{yi}$  and  $U_{yj}$  are Boolean variables. If the relationship  $y$  contains classes  $x_i$  and  $x_j$ ,  $U_{yi} = 1$  and  $U_{yj} = 1$ .  $W_y$  is the static association weight and has the different values according to the different relationship.

The value of the dynamic relationship  $RD_{ij}$  between classes  $x_i$  and  $x_j$  be defined as follows [9, 11, and 13]:

$$RD_{ij} = \sum_{z \in Z} (V_{zi} \cdot V_{zj} \cdot W_z) \quad (2)$$

Where  $Z$  is the set of use cases.  $V_{zi}$  and  $V_{zj}$  are Boolean variables. If the use case  $z$  uses classes  $x_i$  and  $x_j$ ,  $V_{zi} = 1$  and  $V_{zj} = 1$ .  $W_z$  is the weight of the use case  $z$ , which is based on the criticality, frequency or any other considerations.

The total value of relationships between two classes is computed as follows [9, 11]:

$$R_{ij} = RI_s \cdot RS_{ij} + RI_d \cdot RD_{ij} \quad (3)$$

Where  $RI$  is the importance to static and dynamic relationships and  $RI_s + RI_d = 1$ .

### 3.3 Business Component Assembly Based on Rules Library

Application assembly is the second stage of CBSD. It is to build a business application from business components, which are identified from a requirement model using a clustering algorithm. In the domain engineering, business components are organized to implement the realistic business process based on rules in the domain rule library [10, 14].

Business process can be represented formally as follows [10, 15]:

$$BP = \{\text{Name, Business component, Rule}\}$$

Where Name is the business process's name, Business component = (Name, Function, Code, Input, Output), and Rule = (Rule name, Key-point, Algorithm).

Key-points are the business components that are determinative to the execution of business process. Business component is triggered by key-point of the business process and organized by algorithm to build the realistic business process.

In the business component assembly based on rules library, the algorithm is defined by the associated set between the business components as follows [10, 15]:

$$\Omega = \{ \rightarrow, \wedge, \vee \}$$

Where “ $\rightarrow$ ” is sequential relationship, “ $\wedge$ ” is parallel relationship and “ $\vee$ ” is choice relationship. As shown in Table 1, the associated operations, such as And-split, And-join, Or-split and Or-join, can be described with the above three relationships.

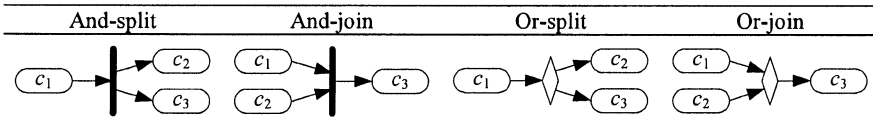
And-split:  $c_1 \rightarrow (c_2 \wedge c_3)$

And-join:  $(c_1 \wedge c_2) \rightarrow c_3$

Or-split:  $c_1 \rightarrow (c_2 \vee c_3)$

Or-join:  $(c_1 \vee c_2) \rightarrow c_3$

**Table 1.** Associated Operations of Business Components



Business rules are described in ECA (Event-Condition-Action) mechanism. The event is the reason of action, condition is the constraint of action, and action is an execution process of operation. Universally, business rules can be described as follows [15]:

```
rule <rule name> [(<parameter>, ...)]
when <event expression>
if <condition i> then <action i>
end-rule [(rule name)]
```

Rule library consists of meta-data set and rule set [10]. Meta-data set describes rule name, meaning, origin, format, purpose and so on. It provides the method of rule access and search. Rule set is a set of domain business rules to describe the content of rules. It consists of business key-point set and algorithm set. Key-point set is a sequential list set of business key-point in business process. Algorithm set is a sequential list set of operator corresponding key-point. As shown in Fig. 2, while the management-scheduling center calls a rule, a key-point list and corresponding several operators are called simultaneously [10, 15].

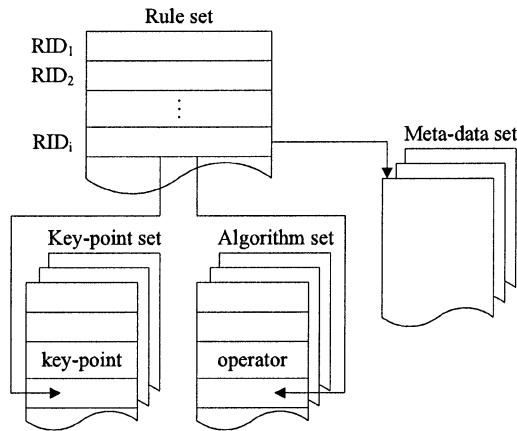


Fig. 2. Business rules library

#### 4 Business Component-Based Development of Enterprise Information System

Development of enterprise information system is to build a business application using business component-based approach. The architecture is a crucial product in software development and can provide guidelines and supports for business components design, including identification and assembly. It is also the center of the component-based software development [10].

At present, hierarchy architecture is one of the universal architectures of enterprise information system. It places the different characteristics of system into the different layers of architecture and each layer provides a service to the layer above and serves as a client to the layer below [16]. Based on the existing references [10, 11, and 16], component-based hierarchy architecture of enterprise information system has five layers from bottom to up, which are software infrastructure layer, middleware container layer, business entity layer, business process layer and application interface layer, as shown in Fig. 3.

The software infrastructure layer consists of operating system (OS), data base management system (DBMS) and network and provides a basic environment on which the whole information system runs. The middleware container layer offers a software-bus mechanism and the business components can plug onto the software-bus to realize plug & play. The business entity layer contains the entity components (EC), which are the encapsulation of business entity and business rule. The business process layer contains the process components (PC), which are the encapsulation of business process and business rule. The application interface layer contains the interface components to meet user interface requirement and trigger business event

of business process. The various enterprise information systems can be developed and reconfigured on the basis of the above component-based hierarchy architecture.

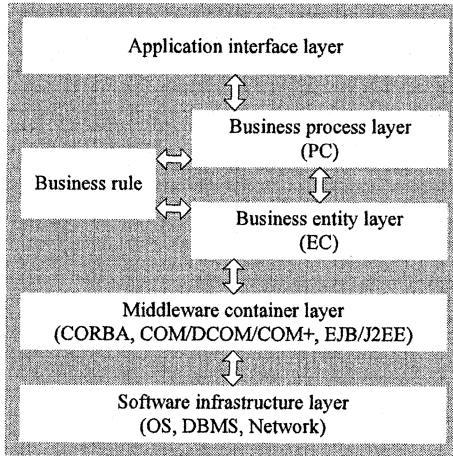


Fig. 3. Component-based hierarchy architecture of enterprise information system

## 5 Conclusions

In order to meet the requirement of business environment changes, enterprise information system must have the characteristics of reconfigurability, reusability and extensibility. In this paper, business component-based approach is used to develop enterprise information system. The business component-based software development process contains two main stages: component fabrication and application assembly. The former is to build the business components using a clustering algorithm and the latter is to build a business application from components using a rules library. The component-based hierarchy architecture is established for development of enterprise information system. With the rapid development and maturation of the standard of middleware such as CORBA, DCOM and EJB, business component-based software development will have been widely applied to develop the reconfigurable enterprise information system.

## References

1. T. Petros, A DSS Model That Aligns Business Strategy and Business Structure with Advanced Information Technology: A Case Study, *Annals of Cases on Information Technology* 6, 157-175 (2004).



2. L. Jyhjong and L. Ming-Chang, An Object-Oriented Analysis Method for Customer Relationship Management Information Systems, *Information and Software Technology* **46**(7), 433-443 (2004).
3. L. Hans, An Object-Oriented Architecture Model for International Information Systems, *Journal of Global Information Systems* **11**(3), 1-18 (2003).
4. Y. Guo, G. Zhang, L. Xie, and Y. Xu, The Research and Design of Business Component Reuse in Enterprise Information System Integration, In the Proceedings of the third IEEE International Conference of Information Technology and Applications (ICITA'05), edited by X. He, T. Hintz, M. Piccardi, Q. Wu, M. Huang, and D. Tien (IEEE, New York, 2005), pp. 41-44.
5. S. Archint and D.N. Batanov, Development of Industrial Information Systems on the Web Using Business Components, *Computer in Industry* **50**(2), 231-250 (2003).
6. P. Herzum and O. Sims (Translated by H. Ke), *Business component factory: a comprehensive overview of component-based development for the enterprise* (China Machine Press, Beijing, 2005).
7. J.A. Kim, Y-T. Jin, and S-M. Hwang, A Business Component Approach for Supporting the Variability of the Business Strategies and Rules, *Lecture Notes in Computer Science* **3482**, 846-857 (2005).
8. K. Wojtek, Architecture Framework for Business Components, In the Proceedings of the Fifth IEEE International Conference on Software Reuse, edited by P. Devanbu and J.S. Poulin (IEEE, New York, 1998), pp. 300-307.
9. H. Jain, N. Chalimeda, N. Ivaturi, and B. Reddy, Business Component Identification: A Formal Approach, In the Proceedings of the Fifth IEEE International Conference on Enterprise Distributed Object Computing, edited by E. Lupu and A. Wegmann (IEEE, New York, 2001), pp. 183-187.
10. X. Li, *Research on the Model of Business Component Oriented Reconfigurable Information System*, PhD Thesis, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 2002.
11. W. Xu, B. Yin, and Z. Li, Research on the Business Component Design of Enterprise Information System, *Journal of Software* **14**(7), 1213-1220 (2003).
12. J. Han and M. Kamber (Translated by M. Fan and X. Meng), *Data Mining: Concepts and Techniques* (China Machine Press, Beijing, 2003).
13. F. Meng, D. Zhan, and X. Xu, Business Component Identification of Enterprise Information System: A Hierarchical Clustering Method, In the Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE'05), edited by S. Ceballos (IEEE, New York, 2005), pp. 473-480.
14. S. Castano and V. De Antonellis, Engineering a Library of Reusable Conceptual Components, *Information and Software Technology* **39**(2), 65-76 (1997).
15. H. Xu, X. Li, and Q. Ding, Business Component Reengineering Based on Rule Library, *Computer Integrated Manufacturing Systems* **9**(10), 911-915 (2003).
16. X. Li, H. Xu, and Q. Ding, Research of Framework of Rapidly Reconfigurable Information System, *Transactions of Nanjing University of Aeronautics & Astronautics* **19**(1), 103-107 (2002).