Intelligent Information Management, **2010, 2, 278-285**
doi:10.4236/iim.2010.23032 Published Online April 2010 (http://www. SciRP.org/journal/iim)

Component-Based Software Development Framework for 3rd Party Logistics Business

Yang-Ja Jang[1], Taehan Lee[2*], Seungkil Lim[3]
[1]*R&D Center, LG CNS, Seoul, Korea*
[2]*Department of Industrial & Information Systems Engineering, Chonbuk National University, jeonju, Korea*
[3]*Division of e-businessIT, Sungkyul University, Anyang, Korea*
E-mail: *bighandy@gmail.com, myth0789@jbnu.ac.kr, seungkil.lim@gmail.com*
Received January 27, 2010*; revised March* 2, 2010*; accepted April* 6, 2010

Abstract

This paper suggests a component-based software development framework for 3rd party logistics (3PL) business. This framework integrates two engineering methodologies in order to identify the most reusable software components that can be used in several types of 3PL business models. UML (Unified Modeling Language) is used to design lower-level software components and DEMO (Design and Engineering Methodology for Organization), one of the business engineering methodologies based on the communication theory, is used to identify core business processes for 3PL business models. By using the methodologies, we develop a 3PL management solution by applying the framework into a C2C type of 3PL business model, specifically the door-to-door (D2D) service.

Keywords: 3PL, DEMO, UML, Software Development

1. Introduction

Intensified business competition has led industries to outsource almost all of business functions except their core competencies. Logistics function is one of the most outsourced one since this function requires tremendous physical investment and considered as non-value added function. Much has been written in recent years about outsourcing logistics activities. There have been various terms used to describe this phenomenon such as logistics alliance, operational alliances in logistics, contract logistics, contract distribution and third party logistics. However, third party logistics (3PL) has been the term more widely used in recent times. Given the growing importance of logistics outsourcing, the logistics service providers (LSP's) have appeared and specialized their services through differentiation, with the scope of services encompassing a variety of options ranging from limited services (for example transportation) to broad ones covering entire supply chain.

Leading LSP's have developed their own logistics management solution themselves and use it on their businesses, while small- and medium-sized LSP's in general buy and use commercial logistics management solutions from professional software vendors for better customer service. Since LSP's provide several types of logistics services to business and/or individual customers, the logistics management solutions should have functionalities to support the services. Generally, 3PL services (in other words 3PL business models) can be classified into three types according to entities involved in the logistics services. B2B type of 3PL services are those of services provided by LSP's for logistics requirements between two business customers. On the other hand, B2C-type 3PL services are performed for logistics requirements between business customer and individual customer (end consumer). Finally, there exist C2C-type 3PL services between individual customers. Although there are several types of 3PL business models provided by LSP's, these models can be supported with a few common business processes. For example, requests for logistics service and acknowledgements of service completion are always needed and performed by LSP's regardless of service types. As a result, we can develop logistics management solutions more effectively and efficiently if we identify the common business processes and we develop software components for the identified business processes since the software components can be reused for several types of services that are performed by doing the common business processes.

In this paper, we suggest a component-based software development framework for 3rd party logistics (3PL) business. This framework integrates two engineering methodologies in order to identify the most reusable software components that can be used in several types of 3PL business models. UML (Unified Modeling Language), known as de facto standard for object-oriented software design, is used to design lower-level software components. DEMO (Design and Engineering Methodology for Organization), one of the business engineering methodologies based on the communication theory [1,2], is used to identify core business processes for 3PL business models. We develop a 3PL management solution by applying the framework into a C2C type of 3PL business model, specifically the door-to-door (D2D) service.

2. 3rd Party Logistics Management Solution Structure

3rd Party Logistics Management Solution is the software needed by 3PL for planning, executing, and controlling the logistics activities. The solution can be categorized by the levels of component granularity within the business component approach. The categories are addressed in order, from finest-grained to most coarse-grained. They are as follows:

1) Distributed component: It has a well- defined runtime interfaces and a runtime network address. Also, it has a specific internal structure into which those object-oriented programming language classes fit or plug.

2) Business component: A component that implements a single autonomous business concept, vendor as an example. It usually consists of one or more distributed components that together address the various aspects of distribution required by the business component.

3) System-level component: A group of business components that cooperate to deliver the cohesive set of functionality required by a specific business need. When a business component system is encapsulated by being provided with clean interfaces designed so that the system as a whole can be treated as a black box, then it becomes a component in its own right: for example, as a vendor management system [3].

Figure 1 shows the relationship between the three different level components. We propose the development process of the logistics management solution component in the next section.

3. Solution Development Process

A business model can be defined as architecture for product, service, and information flow, including a description of business players, their roles, and revenue sources. For example, some of the most popular revenue-generating models adopted by companies are e-tail

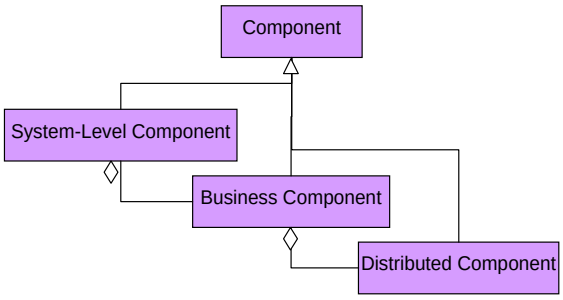

Figure 1. Solution component taxonomy.

model (online retailing) and online auction. The logistics business model also can be defined as a subset of general business models and it has changed very quickly in conformity with the velocity of the logistics business change. As information technology has to support the specific business model, we designate the solution development process in sequence from the business modeling to component system development. **Figure 2** is a development process flow diagram.

3.1. Business Modeling

When developing the logistics management system, an important activity is to design a business model. The purpose of a business model is to describe the fundamental business aspects of the system to be built. Thus, a

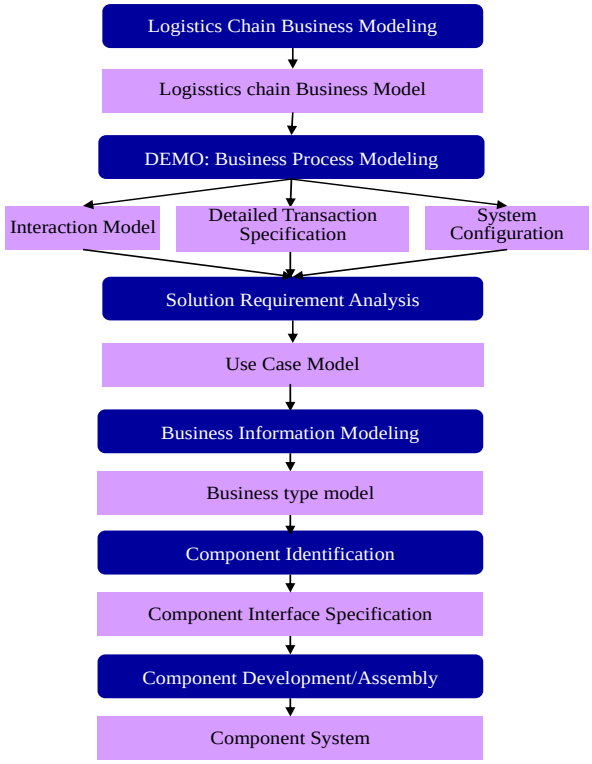

Figure 2. Solution development process.

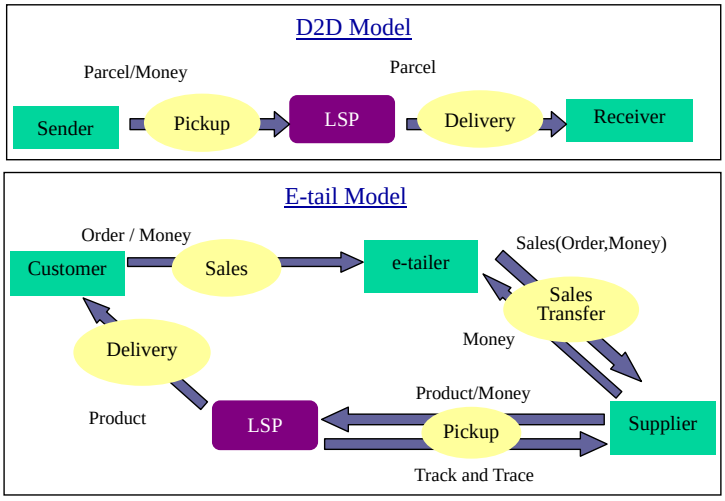

Figure 3. Business model representation.

business model describes which actors are involved, what the actors offer each other, and what activities they perform [4]. The central concept in a business model is that of value, and the model described how value is exchanged between actors in a network view. **Figure 3** describes the door-to-door (D2D) logistics business model and the e-tail model.

Following Gordijn *et al.* [4], we identity the following basic notions of a business model:

- Actor (A)-A set of actor types
- Value Object (VO) -A set of value object types. A value object is a service or a thing that is of value to one or more actors
- Value Transfer (VT) -A set of value transfer types. A value transfer is the transfer of a value object from one actor to another actor.
- Value Exchange (VE) -A set of value exchange types. value exchange consists of two value transfers. The intuition is that a value exchange consists of two reciprocal acts one actor providing another actor with something of value and receiving something of value in return.

In a D2D model, a sender and LSP exchanges a parcel and money and the LSP transfers the parcel to a receiver.

3.2. Business Process Modeling

DEMO (Dynamic Essential Modeling of Organization) is a business process modeling methodology developed by Reijswoud and Dietz, in Delft University of Technology [1,2]. This methodology offers a conceptual framework that is suited as a starting point and basis for modeling information systems and or process modeling [5-8]. The theoretical foundation of DEMO is called the Order-Execution-Result-paradigm, OER paradigm in short. This paradigm basically states that it is possible to acquire an understanding of an organization that is fully abstracted from any kind of realization. What is left of the knowledge about an organization after separating out all realization aspects is called its essence, *i.e.*, the essential business process for the organization to conduct its own business.

In a D2D logistics business model, we identify four essential interaction business processes with the external actors and twelve internal business processes as shown in **Figure 4**. For example, business process T04 can be defined as "Pickup person: requests: Customer: Parcel fee <PO> is paid" in OER notation.

In the course of process modeling, the detailed information transaction and data manipulation needs can be specified in the table format shown in **Table 1**. It is used for the information modeling in the conceptual manner.

Then, we reflect the system boundary specified by the business process model and the essential internal business processes in DEMO to identify the system-level components shown in **Figure 5**. The system consists of four modules categorized by function and each module consists of several system-level components.

3.3. Business Process Modeling

The Unified Modeling Language (UML) is becoming a de facto standard language for modeling automated information systems. UML offers a standard way to write a system blueprint including conceptual things such as business processes. Use Case model describes a set of sequences of actions, including variants that a system performs that yields an observable result of value to an actor. It delimits the system boundary, identifies all relevant system elements or parts, and specifies system use cases.

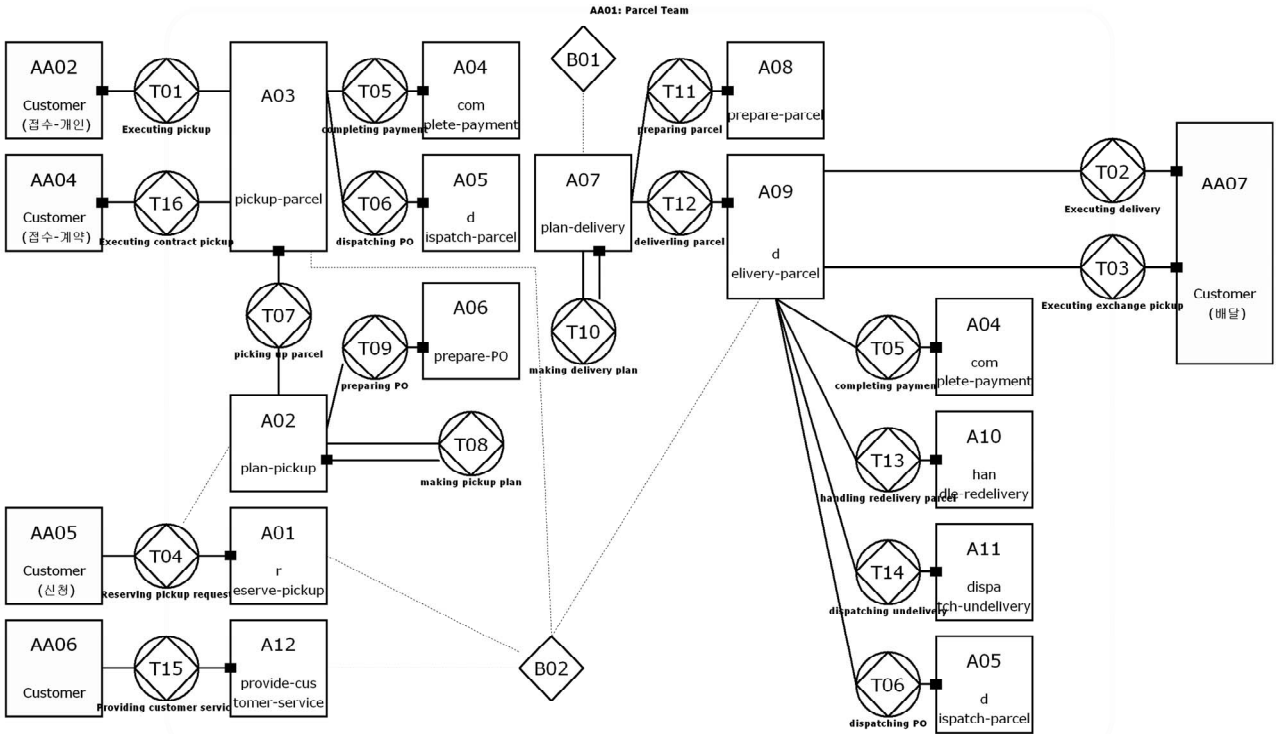

Figure 4. Business process model of D2D.

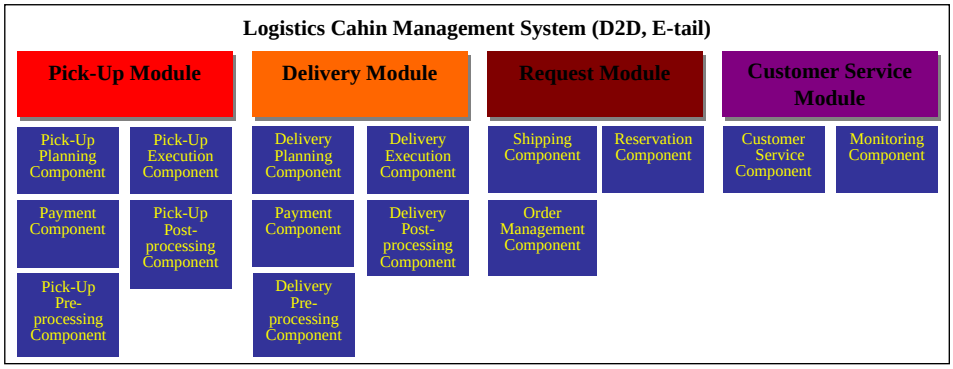

Figure 5. Logistics management system component for D2D.

3.4. Business Information Modeling

The business process description introduces a number of terms, such as reservation, order, and customer, which we need to get clear about. We can construct a mind map that relates these and other important terms, business concept. And we identify business types by converting the business concept. Business types have their unique business identities and they are independent information objects not having enforcing relationships with other objects [9].

The business type model is represented in **Figure 6** by a UML class diagram which contains the specific business information that must be held being specified.

3.5. Component Identification

For each use case, we consider whether or not there are system responsibilities that must be modeled for each use case steps. If so, we represent them as one or more operations of the appropriate system interface. This gives us an initial set of interfaces and operations of the process component. Next, we decide which types in the business type model we consider to be core. The purpose of identifying core types is to start thinking about which information is dependent on which other information,

Table 1. Detailed data and information transaction specification for business process T4.

Level	Initiator		Executor	Result
E	Customer	requests	post office	Pickup-request <PO> is reserved
I	Customer	requests	enter-pickup-info(call center)	Pickup-request-info <PO> is received
I	Customer	requests	manage-pickup-fee(call center)	Pickup-fee-request <PO> is received
I	manage-pickup-fee (call center)	requests	compute-pickup-fee(call center)	Pickup-fee-request <PO> is received
I	compute-pickup-fee (call center)	requests	receive-pickup-fee(call center)	Estimated-pickup-fee <PO> is received
I	receive-pickup-fee (call center)	requests	Customer	Estimated-pickup-fee <PO> is received
I	Customer	requests	manage-available-time(call center)	Pickup-available-time-request <PO> is received
I	manage-available-time (call center)	requests	compute-available-time(call center)	Pickup-available-time-request <PO> is received
I	compute-available-time (call center)	requests	receive-pickup-available(call center)	Available-time-info <PO> is received
I	manage-pickup-available (call center)	requests	Customer	Available-time-info <PO> is received
I	Customer	requests	manage-pickup-reservation(call center)	Selected-time-info <PO> is received
I	manage-pickup-reservation (call center)	requests	Customer	Pickup-reservation-info <PO> is received
I	manage-pickup-reservation (call center)	requests	manage-store-pickup-reservation(call center)	Pickup-reservation-info <PO> is received
D	manage-store-pickup-reservation (call center)	requests	store-pickup-reservation(call center)	Pickup-reservation-info <PO> is stored
D	manage-pickup-reservatio n (call center)	requests	PLIIS	Pickup-reservation-info <PO> is transported
D	manage-pickup-reservation (call center)	requests	pickup-parcel	Pickup-reservation-info <PO> is transported
I	Customer	requests	manage-cancel-reservation(call center)	Reservation-cancel-info-request <PO> is received
I	manage-cancel-reservation (call center)	requests	Customer	Reservation-cancel-info <PO> is received
I	manage-cancel-reservation (call center)	requests	manage-store-cancel-reservation(call center)	Reservation-cancel-info <PO> is received
D	manage-store-cancel-reservation (call center)	requests	store-cancel-reservation(call center)	Reservation-cancel-info <PO> is stored

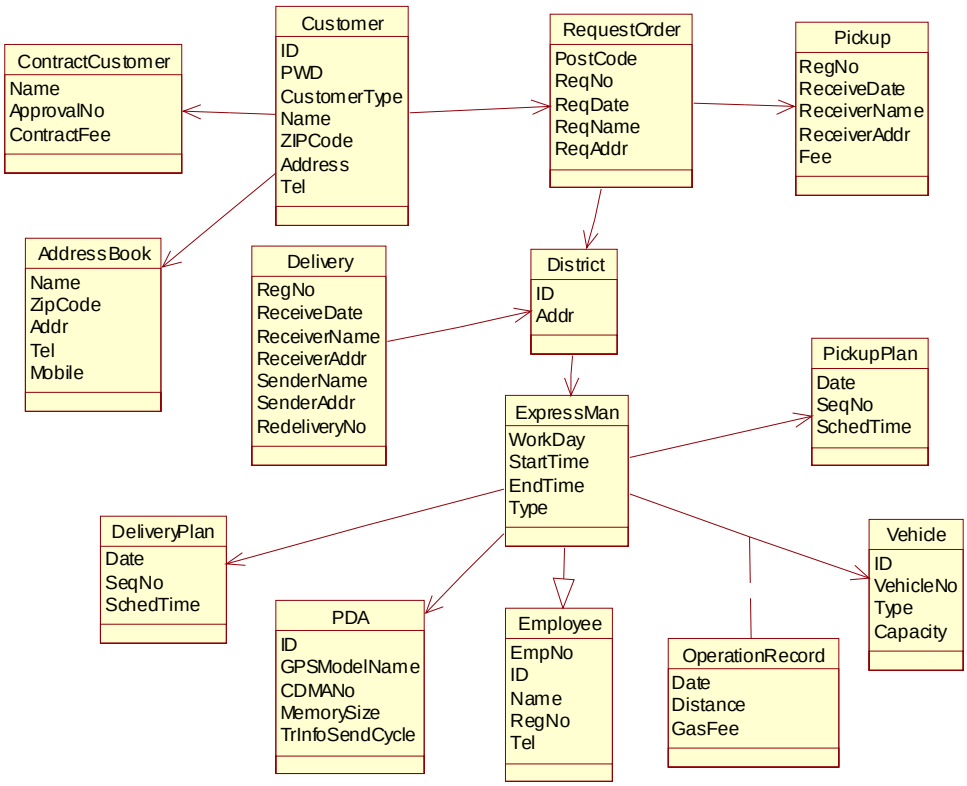

Figure 6. Business type model.

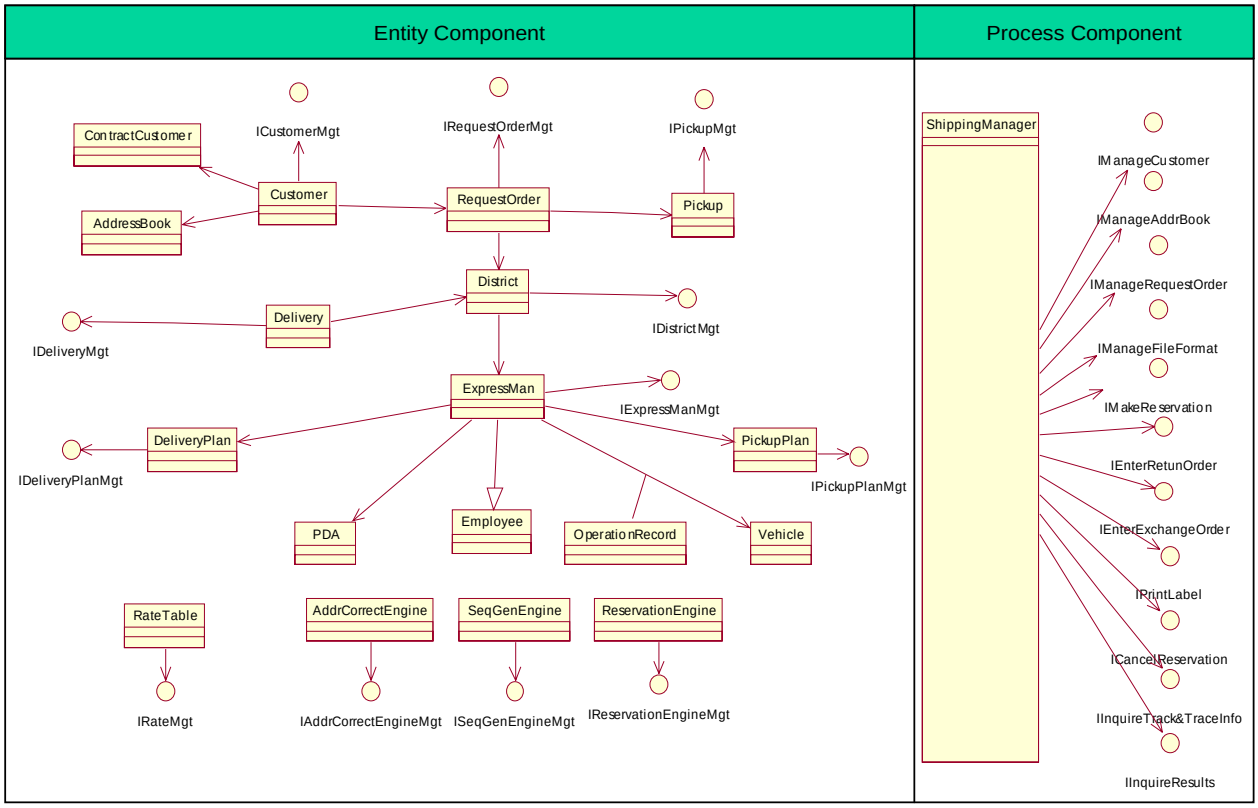

Figure 7. Component specification diagram.

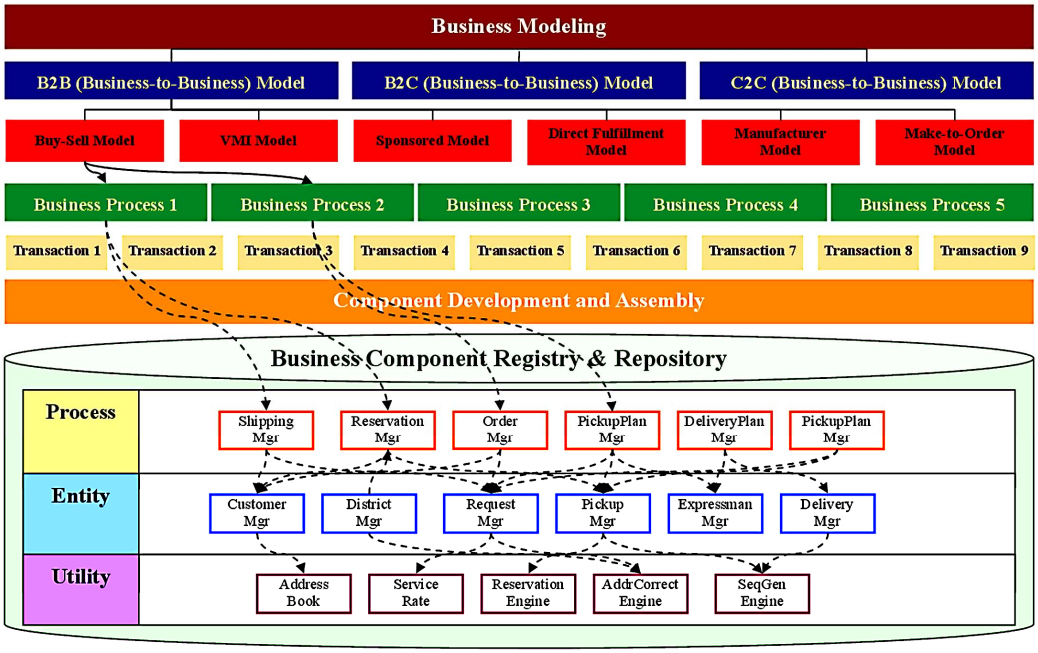

Figure 8. Component system development framework.

and which information can stand alone [9]. Generally, we create one business interface for each core type in the business type model. Each business interface manages the information represented by the core type. We call these entity components. The process component and entity component is described in **Figure 7**.

3.6. Component Development and Assembly

Component systems adhere to the principle of divide and conquer for managing complexity and component reusability. Component reusability does not mean that the-same component can be reused in any other systems but also indicates that the component can be replaced by another component with the same interface specification as the business process changes. We propose the inter-linked hierarchical component system development framework from the business modeling, in which the process modeling is the core process, to the component development and assembly. The framework is shown in **Figure 8**. A new or reengineered system can be assembled into by business components from the business-component registry and repository. Also new or revised components may be managed in the component registry and repository.

4. Concluding Remarks

Logistics business is expected to be developed into a logistics web stage, in which the LSP's are partners collaborating for business effectiveness and efficiency. In this stage, the logistics business model can be generated, changed, united and disappeared more quickly than now. Being in collusion with this trend, component based development methodology for logistics business is required to support the various business models.

This study has a significance to suggest the framework to examine the components to support the business model and can be used to develop the multi-stage or web-stage logistics management solution.

6. References

[1] V. E. van Reijswoud and J. L. G. Dietz, "DEMO Modeling Handbook," Vol. 1, Department of Information Systems, Delft University of Technology, 1999.

[2] J. L. G. Dietz and H. B. F. Mulder, "Realizing Strategic Reengineering Objectives with DEMO," Proceedings of the International Symposium on Business Process Modeling, 1996.

[3] P. Herzum and O. Sims, "Business Component Factory: A Comprehensive Overview of Component-Based Development for the Enterprise," OMG Press, 2000.

[4] J. Gordijn, J. M. Akkermans and J. C. Vliet, "Business Modeling is not Process Modeling," Proceeding of eCOM2000 workshop in 19th International Conference on Conceptual Modeling, 2000.

[5] P. Jayaweera, P. Johannesson and P. Wohed, "From Business Model to Process Pattern in e-commerce," Proceedings of the Sixth International Workshop on the Language-Action Perspective on Communication Modeling, 1999.

[6] J. Barjis, "Automatic business process analysis and simulation based on DEMO," Enterprise Information Systems, Vol. 1, No. 4, 2007, pp. 365-381.

[7] J. L. G. Dietz, "The Deep Structure of Business Processes," Communications of the ACM, Vol. 49, No. 5, 2006.

[8] P. Green and M. Rosemann, "Integrated Process modeling: An Ontological Evaluation, Information Systems," Vol. 25, No. 2, 2000, pp. 73-87.

[9] J. Cheesman and J. Daniels, "UML Components: A Simple Process for Specifying Component-Based Software," Addison-Wesley, 2001.

