

# Multi-view Ontology Based Logistical Management System

Dr. Wen-Ching Liou, Department of MIS, National Chengchi University, Taiwan  
Jiing-Yao Chang, Department of MIS, National Chengchi University, Taiwan

## ABSTRACT

In this paper, we conduct the multi-view ontology approach to develop knowledge-based logistical management system for lead logistics provider also known as fourth-party logistics (4PL). We believe that sharing the consolidated and elaborated domain knowledge between service providers and customers can achieve the win-win situation. By means of conceptual abstraction, 4PL can establish semantic-consistent domain ontology across industries to alleviate the risk associated with disagreement over the interpretation. The application framework extracted from existing e-logistic application can make 4PL more competitive. In addition, we adopted design patterns and provided flexible interfaces that would allow customer to consume external services comfortably for response to environmental changes. Therefore, 4PL can easily customize an instance of our application framework that will boost his e-logistic platform to get better support to knowledge-based logistical management and higher integrity with other business applications.

**Keywords:** *Logistical Management System, Ontology, Application Framework, Design Science.*

## INTRODUCTION

In modern global trade environment, logistics as a core competency aims at satisfying customer requirements by efficient operations at the lowest total cost. In further, firms that obtain a strategic advantage based on logistical competency establish the nature of their industry's competition (Bowersox & Closs, cop. 1996). However, not all companies can afford logistics expertise or asset in-house especially in the emerging States such as Southeast Asia and China. Fourth-party logistics (4PL) was introduced as logistics services provider which integrates the resources, capabilities, and technology of its own organization and other organizations to design, build and run comprehensive supply chain solutions (Wikipedia contributors, 2008 ).

As the 4PL model is providing outsourced logistics services in essence, it is inevitable to face the risks associated with loss of control, lack of expertise in-house, head-butting and long-term partnerships (Richardson, March, 2005). We believe that the 4PL providers will obtain the vantage point if they have the ability to help their clients mitigating the risks. That is the ability of mastering logistics domain knowledge not only in generic case but also in specialized instance. First of all, domain knowledge must be represented under the same conceptual schema (a.k.a. ontology) so as to mitigate the risk of inconsistency. The well-developed abstracted processes and application framework can rapidly respond to client's demand or environmental change. At least but not last, a proper logistics knowledge portal is an effective recipe for making clients gain the power of logistics knowledge and pulling in the potential customers.

In this paper, we propose the multi-view ontology approach to develop an application framework of processing logistical performance cycles which support procurement, manufacturing and physical distribution. To integrate the explanation of logistics terms in diverse industries, we abstract the domain concept in standard knowledge representation language to satisfy the requirement of semantic interoperability. Then, we will show that it is able to use the proposed architecture to get the benefit of knowledge-based management system and applying ontology on logistics.

In the next section, we will briefly introduce the ontology, as well as the introduction of object-oriented application framework. In section of methodology, we will illustrate the research methodology and our multi-view ontology model. Our application framework for logistical management will be discussed in subsequent section. At the end we draw a conclusion of our work.

## BACKGROUND

### Ontology

Ontology research has become relevant to construct knowledge-based systems since the beginning of 1990. One of the most popular definitions of ontology considers that ontology is a formal, explicit specification of a shared conceptualization (Borst, 1997). From this point forward, ontologies aim to capture consensual knowledge in a generic way and consolidate the explicit defined machine-readable terms, concepts, properties, relations, functions, constraints and axioms of some phenomenon in the world to form the abstract model which can be reused and shared across software applications and by groups of people.

There are several types of ontologies distinguished from different criteria. In terms of richness levels of description, they can be categorized to lightweight and heavyweight ontologies respectively. The lightweight ontologies describe the terms, concepts, concept taxonomies, properties and relationships between concepts of the target. The heavyweight ontologies, on the other hand, can be regarded as the expanded lightweight ontologies on constraints and axioms (Gomez-Perez, Fernandez-Lopez, & Corcho, 2004). According to the generality of ontologies, there are four types of ontologies (Studer, Benjamins, & Fensel, 1998):

- Application ontologies – they capture the knowledge required for a particular application and are application-dependent.
- Domain ontologies – they specify the knowledge of a particular type of domain and can be reusable in a given specific domain.
- Generic ontologies – they contain common sense knowledge which are valid across several domains.
- Representational ontologies – they are domain-independent and describe the representation primitives used to model knowledge under a given knowledge representation paradigm.

Based on the subject of conceptualization, we can distinguish further between static structural and dynamic behavioral ontologies. These types of ontologies are not only used for classifying, but also practically modeling the real-world phenomenon from different views. We'll illustrate the idea of multi-view ontology approach to develop knowledge-based management systems in the methodology section.

### Application Framework

In order to make the system can adapt to different organization or industrial environments after the simple customization, we adopt the technique of object-oriented application framework development to benefit from modularity, reusability, extensibility, simplicity and maintainability (Chen, 2004). The framework is a “semi-finished” application which is a reusable design of all or part of a system that is represented by a set of abstract classes and the way their instances interact (M. Fayad & Johnson, 2000). An consolidate application framework should describe component objects and how they interact, the interface of each object and the flow of control between them, how the system's responsibilities are mapped onto its objects, how a system is divided into its components and how to reuse the high-level design of a system (M. Fayad, Schmidt, & Johnson, 1999). These design elements should be consider as the criteria of completeness to judge the quality of application framework.

According to instances producing mechanism, there are three kinds of application framework:

- Black-box frameworks consist of concrete and ready-to-use classes. Adapting a black-box framework allows developers to assemble a number of components without looking at their implementation to create the desired result.
- White-box frameworks are made up with a set of abstract classes. Developers rely on inheritance and require more knowledge about business domain.
- Grey-box frameworks consist of both abstract classes and concrete classes. Developers of grey-box frameworks should take both inheritance and composition approach to instantiate an application. The proposed framework in this paper is a kind of grey-box framework.

The framework design is an iteration process. It starts from collecting a number of examples to do domain analysis. Next, architect designs the framework of analyzed domain. The developer can then build applications based on

the framework. After evaluating the result, architect may receive the suggestions to improve the framework. This process may continue through several iterations until the framework is good enough to publish. In design phrase of framework development, Design patterns are massively used inside application frameworks to solve design problems. So, design patterns can be treated as the architectural elements of frameworks.

## METHODOLOGY

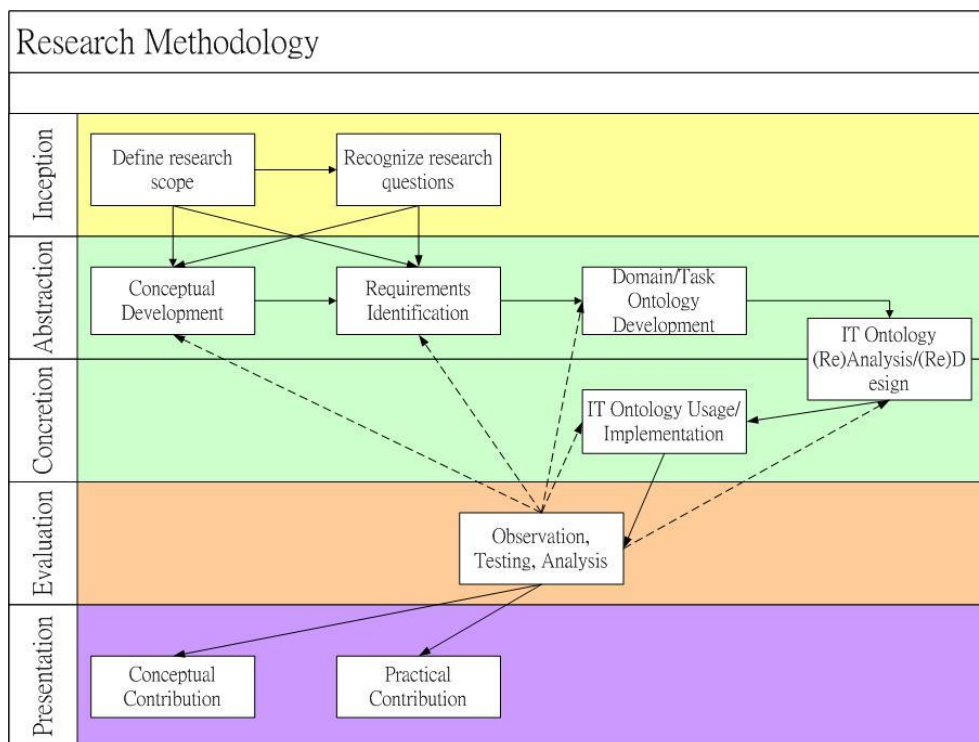
### Research Methodology

The design science is technology-oriented and is attempting to create things that serve human. The application framework development belongs to the design science. Design science usually includes two kinds of activities: Build and Evaluate. Building activity is the procedure of creating artifact which is for some specific purposes, and evaluating activity is the procedure to determine if the artifact conforms to its purposes or not. There are four types of research outputs in design science (March & Smith, 1995):

- Constructs: conceptual elements in research domain.
- Models: relationships among constructs.
- Methods: the guideline or algorithm used to perform tasks on a set of underlying constructs and models.
- Instantiations: the realization of the models.

Furthermore, the detail of the research process of IT system development, that also is a kind of design science research, is illustrated by Nunamaker ,etc. They proposed five major activities in IT system development: Constructing a Conceptual Framework, Developing a System Architecture, Analyzing & Designing the System, Building the Prototype System and Observing & Evaluating the System (Nunamaker, Chen, & Purdin, 1990-91). In addition, Morrison and George drew the whole picture of IT system development based on the process of Nunamaker. Their process is added the stages of research problems/questions identification and conceptual/practical contributions (Morrison & George, 1995).

Consequently, we adopt the research methodology which merges some characteristics of application framework development with the previous study, as in Figure 1.

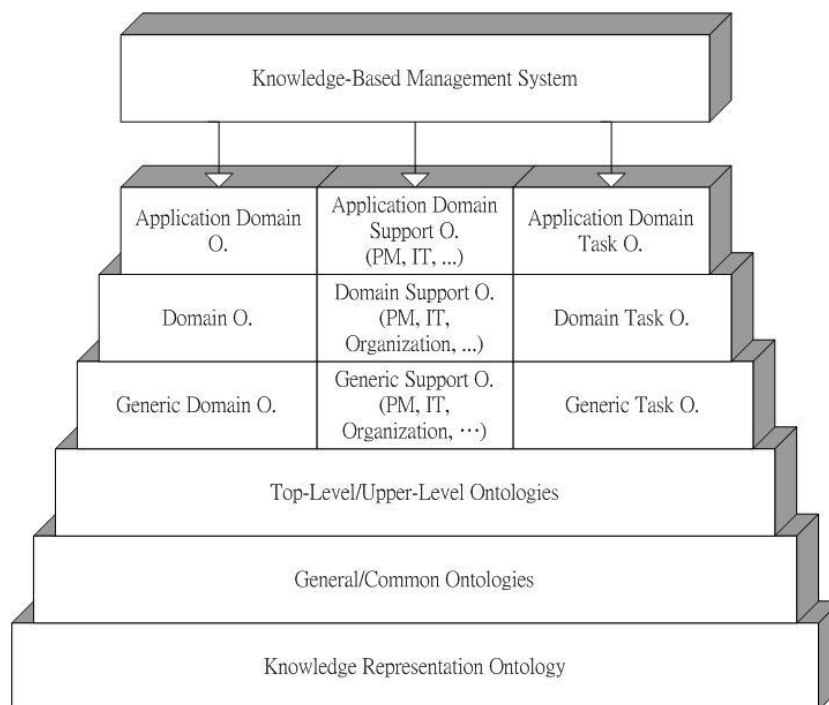


**Figure 1: research methodology for application framework development**

We segment the process into five stages. They are inception, abstraction, concretion, evaluation and presentation, respectively. The abstraction step and the concretion step are build activity of design science research, and the evaluation is considered as evaluate activity. In our study, we first define the research scope is to develop knowledge-based logistics management system at the Inception stage. We then develop logistics domain related conceptual schema, construct application framework, as well as design abstract and concrete component at the Abstraction stage. At the Concretion stage, we implement an executable instance of application framework. Evaluation results may suggest revision of considerations of the activities at Abstraction and Concretion stages. It could process several iterations to make the artifact conform to research goal. Finally, we announce the contribution at Presentation stage.

### Multi-View Ontology Model

Before showing our work, we would like to discuss the multi-view ontology approach to model the knowledge-based management systems first. In terms of modeling, we believe there are still some types of ontologies playing supporting roles to identify consensual knowledge of such as information technology, human resources and project management, etc. We call them as supporting ontologies which can fill the gap in developing knowledge-based management systems practically. We depict the knowledge-based management system which can be derived from the multiple related ontologies of a particular application in Figure 2.



**Figure 2: the multi-view ontology model**

This model is extended from the reusability-usability trade-off problem applied to ontology field (Klinker, Bhola, Dallemagne, Marques, & McDermott, 1991)(Gomez-Perez et al., 2004). The higher the layer, the more specialized and usable ontology is. The bottom layer of model is the meta-language for the knowledge representation languages, such as RDF, OWL, and UML etc., which is the most general and reusable ontology. In this paper, we draw three ontologies from logistics domain to build our knowledge-based application:

- Application domain ontology: it's a static conceptual schema, which is captured from a given logistical management system for single industry, to describe the structure of specialized concept. It is represented in OWL.
- Application domain task ontology: we describe the dynamic behavior from a given logistical performance cycles in OWL-S.

- IT ontology: we proposed the multiple-layered application framework of knowledge-based logistical management system in UML as following.

### PROPOSED LOGISTICAL MANAGEMENT SYSTEM

Our subject is a 4PL which provides integrated e-logistic solution across industries in China and Taiwan. They have been asked to speed up system implementation when they acquire new customers, as well as the new system should keep semantic consistency for maintenance issues and be adaptable to workflow changes. After the discussing the system scope and problems confronted, we applied multi-view ontology approach to build knowledge-based logistical management system which can be as their future core competitiveness.

#### Application level ontology

The first step of developing ontology, which is advised by most guides, is searching existing related ontology for reusability. There is no proper logistics ontology available on public Internet. Nevertheless, we select IBM's Information Framework (IFW) for concept categorizing. The Financial Service Data Model (FSDM) of IFW defines nine categories of data concepts such as involved party, event, location, product, arrangement, condition, classification, business direction item and resource item (Evernden, 1996). We collected the project documents from file store to establish a glossary. Then, we create concepts, instances and relations from glossary, entity-relationship diagrams and data flow diagrams. The concept of involved party is depicted by VisioOWL for communication purpose in Figure 3. The finalized version of domain ontology is composed by protégé tool in OWL and OWL-S (partially shown as Figure 4).

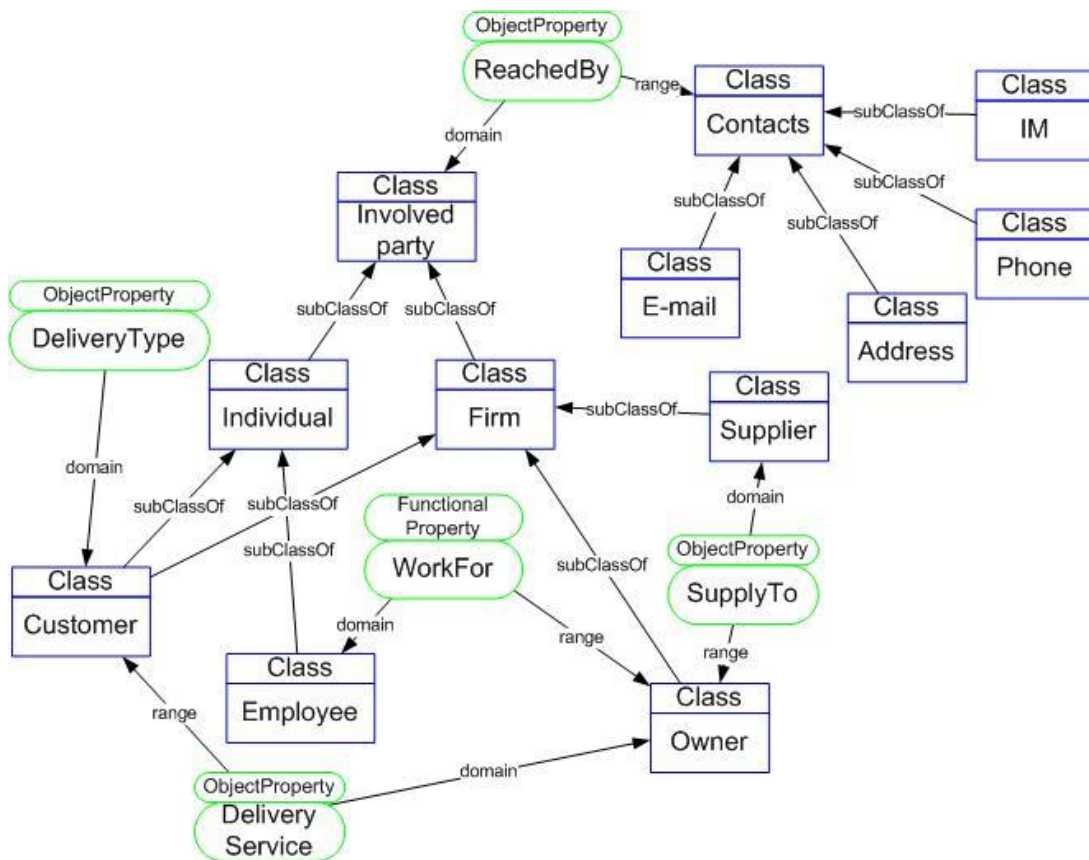
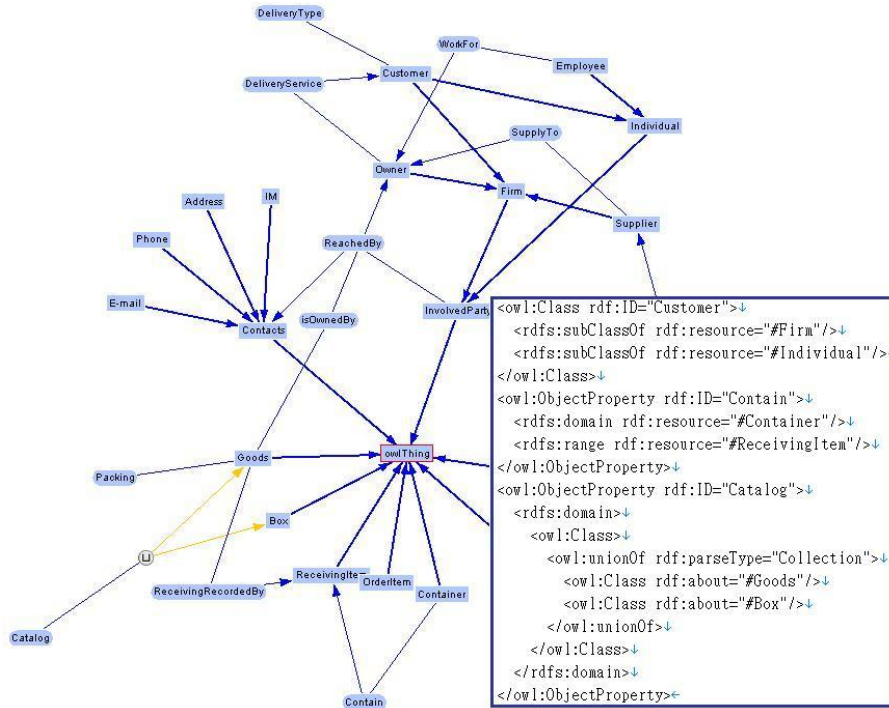


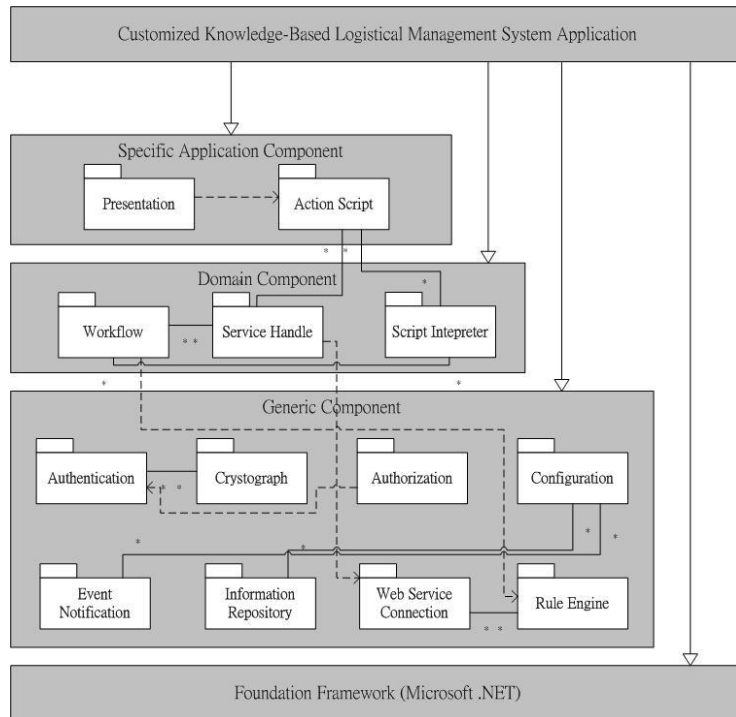
Figure 3: part of application domain ontology



**Figure 4: part of application domain ontology represented in OWL**

### Application framework of knowledge-based logistical management system

Our design is made use of multiple-layer application framework model (Chen, 2004) including four layers in our application framework. As shown in Figure 5, they are (from the bottom): Foundation Framework Layer, Generic Component Layer, Domain Component Layer, and Specific Application Domain Layer. These four layers are not distinguished by the physical architecture or the geographical deployment; they are segmented with the level of business knowledge.



**Figure 5: the application framework of knowledge-based LMS**

Foundation Framework Layer can be treated as the infrastructure of system development. The implementations of multi-threading, garbage collection, and remote procedure calling are included in this layer. It closes by OS level and so is developed by large software vendors, such as Sun, the initiator of Java Environment, and Microsoft maintaining the .NET Framework. In this paper, we adopt .NET framework as the foundation framework because .NET framework provides more supporting on Microsoft family products which our subject is more familiar with. Generic Component Layer contains the common components and services which can be referenced by any domain application, despite their business domain. This layer contains:

- Authentication component serves the process of identifying an individual who attempt to access target system.
- Cryptography component provides the data encryption and decryption service for the application.
- Authorization component serves the process of granting or denying individuals access to a specific underlying resource based on their identity or group.
- Configuration component supports a consolidated service for providing application-specific configuration information at runtime.
- Event Notification is a component providing an event-notification model for any business domain of applications.
- Rule Engine is a component which is able to apply the business rule and then take some action based on the inference of rule.
- Information Repository component provides the storage mechanism to organize and access data easily.
- Web Service Connection component helps to process the communication of web service.

Domain Component Layer, on the contrary, includes the framework components with the logic in a particular business domain. We draw the following components in this layer:

- Workflow component can be considered as a programming model for service components composition and can support process flexibility by application behavior through rule engine component.
- Service Handle component assists application to consume service provided by other system. In order to hook heterogeneous services, we adopt the famous adapter design pattern of GoF, which can convert the interface of adaptee class (i.e., arbitrary systems) into a target interface that client can understand (Gamma, Helm, Johnson, & Vlissides, 1995).
- Script Interpretation component provides the soft-coding script execution at runtime.

Finally, the Specific Application Layer groups the components of:

- Presentation component adopts the mediator pattern that provides a unified interface to a set of interfaces in visualized controls.
- Action Script component makes screen operation be more customizable and extensible.

We can get customized instances derived from application framework that target to a specific application, e.g., we derived an instance named “Bookware Logistical Management System” from the specific application domain of bookstore distribution centers.

## CONCLUSION

The 21st century is the century of the knowledge based economy, most of firms and organizations want to squeeze into the train of knowledge related topic desperately in order to consolidate core competence and pursue excellence. In this paper, we proposed practical multi-view ontology model to construct the application framework of knowledge-based logistical management system. The benefits of our solution would be:

- The terms of logistics will keep semantic consistency across industries, which will facilitate 4PL’s entrance into the markets have not opened up.
- The abstract model can be easily to visualize for communicating precisely.
- The well-developed domain ontology can be leveraged to establish knowledge-sharing mechanism such as knowledge portal or domain-pedia, which will promote the core value and acquire new customers.
- The artifact of IT ontology, i.e. application framework, provides system-wide concept which can accelerate to meet the demand of customers in various industries.

- As to those companies in need of logistics service, they can adopt our model to evaluate the service provider or build its own management system at lower total cost.

We consequently proposed a multiple-layer application framework which can be easily derived to build the customized logistical management system for supporting logistical performance cycle. Depending on the demand of customers, we can also build an instance of framework with features of decoupled workflow, friendly user interface, e-mail-based event notification, adjustable authentication mechanism, external services integration, and so on. Furthermore, Framework is designed on solid design science research methodology to improve quality of result.

Finally, we would like to point out the limitations of this research and the direction of future works. First, the application framework is not suitable for single case because of the longer development cycle and more resources consumption. More instances of application framework can unfold more protruding advantages of application framework, such as reusability, extensibility and maintainability. It is a long journey to get an acceptable achievement, but it's worthy for 4PL provider especially. It may be helpful to release application framework through open source way under the permission of research sponsor, and could be able to improve quality of application framework from the world-wide user's feedback. Second, without the proper standard measurement to evaluate application framework, it cannot objectively judge the quality of an application framework. This problem may combine the survey research on user satisfaction or develop the metric to analyze application framework quantitatively in the future.

## REFERENCES

- Borst, W. N. (1997). Construction of engineering ontologies for knowledge sharing and reuse. University of Twente. *PhD Thesis*
- Bowersox, D. J., & Closs, D. J. (cop. 1996). *Logistical management: The integrated supply chain process*. New York: McGraw-Hill.
- Chen, X. (2004). *Developing application frameworks in .NET* (1st Ed.). USA: Apress.
- Evernden, R., (1996). The information FrameWork. *IBM Systems Journal*, 35(1), 37.
- Fayad, M., Schmidt, D. C., & Johnson, R. E. P. D. (1999). *Building application frameworks: Object-oriented foundations of framework design* (1 edition ed.). Wiley New York N.Y.: John Wiley & Sons.
- Fayad, M., & Johnson, R. E. P. D. (2000). *Domain-specific application frameworks: Frameworks experience by industry*. Wiley New York N.Y.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns* (1st ed.). USA: Addison-Wesley Professional.
- Gomez-Perez, A., Fernandez-Lopez, M., & Corcho, O. (2004). *Ontological engineering : With examples from the areas of knowledge management, e-commerce and the semantic web*. London ; New York: Springer-Verlag.
- Klinker, G., Bhola, C., Dallemagne, G., Marques, D., & McDermott, J. (1991). Usable and reusable programming constructs. *Knowl.Acquis.*, 3(2), 117-135.
- March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15(4), 251-266.
- Morrison, J., & George, J. F. (1995). Exploring the software engineering component in MIS research. *Communications of the ACM*, 38(7), 80-91.
- Nunamaker, J. F., Chen, M., & Purdin, T. D. M. (1990-91). Systems development in information systems research. *Journal of Management Information Systems*, 7(3), 89-106.
- Richardson, H. (March, 2005). *What are you willing to give up?* Retrieved 2008/3/30, 2008, from <http://www.logisticstoday.com/displayStory.asp?S=1&sNO=7028>
- Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1-2), 161-197.
- Wikipedia contributors. *Third-party logistics*. Retrieved 2008/3/30, 2008, from [http://en.wikipedia.org/wiki/third-party\\_logistics?oldid=196720364](http://en.wikipedia.org/wiki/third-party_logistics?oldid=196720364)