A Conceptual Model of ERP for Small and Medium-Size Companies Based on UML

Jae-won Park[†] and Nam-Yong Lee[††],

School of Computing, College of Information Science
Soongsil University, Seoul, Korea

Summary

Over the last decade, numerous companies have tried to adopt a commercial ERP package in the world-wide e-business and e-commerce environment. However, most of commercial ERP packages are designed for a large-scaled company so that it is difficult to adopt a commercial ERP package in terms of small and medium-size companies. Therefore, it is necessary for a small and medium-size company to seek for an approach for the ERP solutions. In this paper, authors describe a conceptual ERP model for small and medium-size companies by using "4+1" views of Unified Modeling Language. The conceptual ERP model consists of five subsystems: Manufacturing, Sales, HumanResource and Payroll, Trading, and Accounting. The conceptual ERP model is an architectural approach to enterprise systems. By using the model, small and medium-size companies, especially manufacturing companies, can afford to achieve global business process and to acquire ERP systems efficiently.

Key words:

ERP, UML, Small and Medium-Size Company, Object-Oriented Technology

Introduction

As an electronic business environment changes more rapidly under the globalization, even small and medium size companies also change their business. With enterprises becoming bigger and bigger, the legacy business systems may not be flexible enough to adapt this change and the discordance between business and information systems in their organization may occur [2]. Therefore, recently most companies use an Enterprise Resource Planning (ERP) system for improving core competency.

There are several definitions of an ERP system. Especially, American Production & Inventory Control Society defines ERP as "an accounting-oriented information system for identifying and planning the enterprise-wide resources needed to take, make, ship, and account for customer orders". Gartner Group describes ERP as "a set of applications designed to bring business functions into balance and represents the next generation of business systems". On the other hand, ERP is a comprehensive packaged software solution that aim for total integration of all business functional areas. Thus, the authors can conclude that an ERP is the generic name of this new class of packaged application software for enterprise integration under electronic business environment.

The term, ERP has been introduced in the early 1990s as the successor to Materials Requirement Planning II, itself a successor to the Materials Requirements Planning software that results from requirements for greater control and efficiency in manufacturing systems [14]. While ERP has its origins in manufacturing and production planning systems, the scope of ERP has expanded in the mid-1990s to include other functions: order management, financial management, production control, quality control, asset management and human resources management. The concept of ERP could be named as "back-office" functions [14]. Recently, the functional scope of ERP systems has further expanded to include various functions such as Electronic Commerce, Supply Chain Management, and Customer Relation Management. The concept of ERP could be named as "front-office" functions [14]. Therefore, ERP may cover all business functional areas and ERP system has became a core business system in the global electronic business environment [24].

Since the 1990s, companies of all sizes and industries have tried to adopt ERP systems in order to improve business processes or replace legacy business systems [14]. But, not all companies achieve their goals by implementing ERP systems because commercial ERP packages are very large and complicate. The implementation of an ERP system is difficult and not simple activities. It involves a complex

set of tasks. For example, nearly half of the large-size companies which adopted ERP systems over the past five years have experienced significant time delays and budget overruns [3]. The ERP system, taking up to large companies, is complex and huge package such as Oracle [25], SAP [5], and Baan [27]. It does not adopt enough for small and medium-size companies because of its size and complication. Small and medium-size companies are by nature more at risk in adopting ERP systems. Because there is a lack of resource, including human resource, budget, and time to devote to implementing ERP systems. T.J. Elliott pointed out that small and medium-size companies have smaller information technology departments and less experience with large-scale projects such as ERP [26]. Therefore, the authors propose a new approach to acquire ERP systems for small and medium-size companies effectively. This approach complies with that ERP systems must be very simple, cheap, compact size and easy to adopt needs of a company, budgets and culture [22].

In this paper, the authors propose a conceptual ERP model for small and medium-size companies by using "4+1" views, based on Unified Modeling Language (UML). This conceptual ERP model may achieve a global business process for small and medium-size companies, especially in manufacturing. The conceptual ERP model can be used to developing ERP systems in a small and medium-size company.

2. Literature Review

The authors have recognized the lack of the literature associated with the conceptual ERP models based on object-oriented technology. Object-oriented technology has been gained attention to overcome software crisis [12]. This means that currently, object-oriented technology can be used to develop business information systems, including ERP systems. Object-oriented modeling implies analysis and design phase by using object-oriented technology [4]. Object-oriented modeling has proven to be an excellent technique for modeling business processes in a company [13]. Recently, business modeling is a new area for object-oriented modeling and has generated a lot of interests.

In general, a model is an abstraction of a system, specifying the modeled system from a certain point and a certain level of abstraction [9]. Modeling a complex system is an extensive and complicate task. Ideally, the authors suppose that entire system can be described in a single diagram. A single diagram clearly defines whole system unambiguously, and is easy to communicate and understand because whole system can be identified at one time [20]. However, it is usually impossible or very difficult to describe overall system in a single diagram because most of business information systems are very large and complicate. Thus, only a single diagram cannot capture all information needed to describe an entire system [20]. When the authors are modeling a system, the system can be described with a number of different aspects: functional, nonfunctional, and organizational. Therefore, ERP systems may be described in several views, which each view represents a projection of the complete system description, showing a particular aspect of the system. In UML, each view is described in a number of diagrams that emphasize a particular aspect of the system [20].

UML is an industry standard modeling language adopted by Object Management Group in 1977. UML is a modeling language intended to describe models of systems – real world and software – based on object concept [15]. Since the goal of UML is to describe any type of systems, UML can be used to model systems, the range of which is very board [13]. UML consists of two vital tools: a notation and a meta-model [18]. The notation is a set of diagramming syntax, which lets you think about and convey your analysis and design. The meta-model is the definition of the notation. UML is rich and complicated notation for describing software systems [18]. Perspectives are views of looking at systems and describe different aspects of user's requirements. In the following section, the authors discuss various views of stakeholders to describe the conceptual ERP model.

3. Research Framework

Similarly building construction, architecture of an ERP system is described as different views of the system being developed. Different views are used to making the important characteristic of system more visible. System architecture is a view of the whole system. System architecture perhaps is the most important artifact that can be used to manage these different viewpoints.

According to UML and Rational Unified Process (RUP), the viewpoint of describing system is based on the "4+1" views. Figure 1 represents the "4+1" views of UML to describe system architecture [8] [20]. Use case View of a system encompasses the use cases that describe the behavior of system as seen by its end users, analysts, and testers. In other words, it describes the functionality the system should provide, as perceived by external actors. And it specifies the forces that shape the system's architecture. In UML, it is captured by use case diagrams
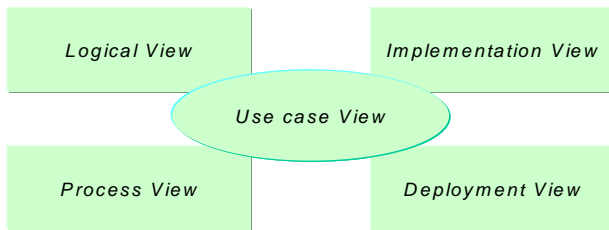
Fig. 1   The "4+1View" Architecture Model

4. Conceptual ERP Model

and occasionally in activity diagrams. Logical View of a system includes classes, interfaces, and collaborations that from the vocabulary of the problem and its solution. It supports the functional requirements of the system. In the UML, it is captured in class diagrams, object diagrams, state diagram, sequence diagrams, and activity diagrams. Implementation View of a system involves the components and files that are used to assemble and release the physical system. It addresses the configuration management of system's releases. In the UML, it is by component diagrams, interaction diagrams, and activity diagrams. Process View of a system consists of thread and processes that form system's concurrency and synchronization mechanisms. It presents the performance, scalability, and throughput of system. In UML, it is captured by the same kinds of diagrams as for the logical view. Deployment View of a system encompasses the nodes that form the system's hardware topology on which the system executes. It describes distribution, delivery, and installation of parts that make up the physical system. In UML, it is captured in deployment diagrams, interaction diagrams, and activity diagrams.

The "4+1" views allow developer to comprehend a complex system in terms of its essential characteristics. In addition to brief explaining each view, it is necessary to remember that the "4+1" views of system architecture are not completely independent [20]: sequence diagram and collaboration diagram represent use case interaction model. Elements of one view are connected to elements in other views.

In the following section, the authors represent results of modeling conceptual ERP system by using "4+1" views. The conceptual ERP model can be described with several diagrams of UML notation. The conceptual ERP model being introduced in the following section depends upon a real situation.

Since the industry of Korea heavily depends on export, trading between countries and manufacturing becomes an important sector. Now, there are many small and medium-size manufacturing companies in Korea and they eagerly do their best for survival in the world-wide electronic business environment. It is more and more difficult for small and medium-size companies to run a business. Because of the competition among the world companies heated up as world economy is opened and business environment becomes more global and rapidly changes. They are not enough prepared with business system and business process redesign for adjusting to the international and radically changed business environments. Thus, many small and medium-size companies, as large-size companies replace their ERP systems, are supposed to adopt a new ERP system which meets with the change of status [2].

However, small and medium-size companies do not have the same organizational structure and a lot of capital which large-size companies do. Commercial ERP systems like Oracle ERP, SAP, and Baan have numerous modules geared to the needs of large-size companies. In addition, it is difficult to resolve the many problems associated with unique requirements such as billing systems because these ERP systems support global standards [25]. There is high risk to apply a commercial ERP package to small and medium-size companies. Therefore, small and medium-size companies need a different approach to ERP systems from large-size companies.

In general, functionalities of ERP systems can be classified into five areas [6]. First, ERP systems provide function to handle hybrid manufacturing. Therefore, we can control the multiplicity of conflicting requirements of users. Second, using by simulation function, ERP systems have the ability of multi-level Master Production Schedule / Material Requirements Planning that eliminates the processing time and multiple scheduling. Third, ERP systems have a high degree of integration capability both with internal system and with external system, business process, and data. Fourth, ERP systems can be running the "multiple" plants concurrently with different operating system. Thus companies can run global business organizations. Finally, ERP systems should assist multi-language operations. Therefore, each user can have the ability to use own national language.

In this section, the authors will introduce a conceptual ERP model for small and medium-size companies based on functionalities of ERP systems. An ERP system consists of five subsystems, including manufacturing,

sales, human resource and payroll, trading and accounting. First, the "Manufacturing" subsystem handles all information with respect with manufacturing: Items, Bill of Materials, Inventories, Purchasing, Product Plans, Production, Outside Production, Equipments and Quality. Term "Item" includes parts, semi-products and products. In the future, it will be explained used by diagrams. Second, the "Sales" subsystem consists of functions: Manage Propose, Manage Order, Manage Contract, Manage Shipping, Manage Returned-Goods, and Manage Customer. Business task for sales may includes the following steps: (1) at first, identify customer whether customer is registered, (2) propose to customer for product, receive order from customer and make a contract with customer for order in detail, (3) ship the goods by contracts, and (4) finally accomplish process of returned-goods. If customer is not identified, sales subsystem registers customer data, and handle data of results, estimation and analysis about customer. Third, the "Human resource and payroll" subsystem provides efficiency for management of human resource information and integrated management of payroll and settlement of accounting. It consists of Manage Information of Human resource, Manage Public Welfare, Manage Payroll, and Manage Settlement of accounting. Fourth, in compare with other ERP system, the "Trading" subsystem is a unique subsystem in the ERP system model. It provides integration and efficiency for the imports and exports processes of company. It includes standard document format for electronic data interchange and has trading information in regarding to international commerce rules. It consists of Manage Buyer, Manage Offer, Manage Order, Manage Letter of Credit, Manage Loading, Manage Entry, and Manage Claim. Finally the "Accounting" subsystem provides efficiency and flexibility for accounting process. It uses integrated Database and keeps correct accounting information. It can be classified into Manage General Accounting, Manage Accounting of Taxation, Manage Budgets and Funds, and Manage Managerial Accounting.

4.1 Use Case View

A use case is typical interactions between user and ERP system and describes functionalities of an ERP system [8][23]. A use case diagram illustrates a set of use cases, actors, and relationships between actors and use cases. The purpose of use case diagram is to show a context of an ERP system. In this section, the authors describe two use case diagrams. One shows splitting up of an entire system into subsystems. The other shows how to describe one of subsystems, "Manufacturing" in detail.

An ERP system is very large and complex package. The authors first divide the entire ERP system to subsystems. Commonly, subsystems may be defined and used to organize a large-scale information systems into smaller and to make it more comprehensible or manageable so that it can be more easily described to other people [21]. As shown in the Figure 2, a conceptual ERP model can be divided into five subsystems: Manufacturing, Sales, HumanResource and Payroll, Trading and Accounting.
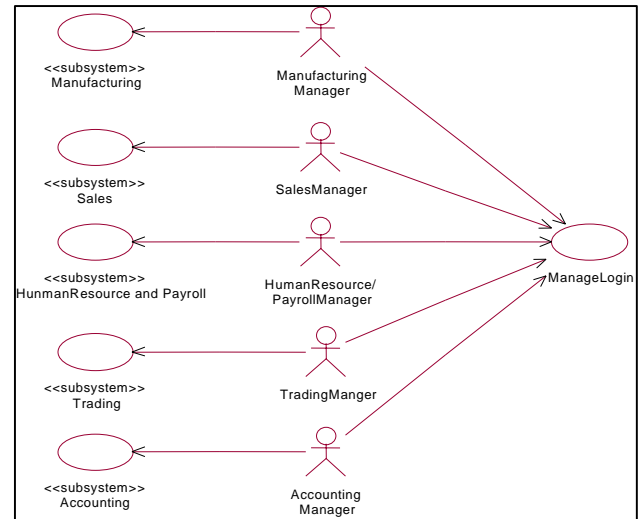

Fig. 2 Use Case Diagram at High Level

In the use case diagram at high level, each use case represents a subsystem except for "ManageLogin" use case. Each actor of use cases, ManufacturingManager, SalesManager, HumanResource/PayrollManager, TradingManager, and AccountingManager, is representative of external person associated with subsystem. For example, when we describe the "Manufacturing" subsystem in detail, the actor "ManufacturingManager" can be divided into Product Control Manager, Purchase Manager, Product Plan Manager, Production Manager and Quality Manager shown in the Figure 3.
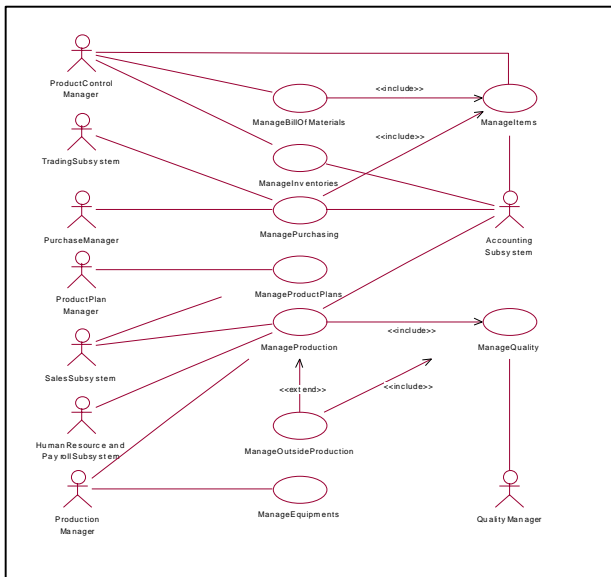
4.2 Logical View

A class diagram describes static view of ERP system in terms of classes and relationships among the classes [8][17]. When we describe the class diagram of "Manufacturing" subsystem, we group classes into nine class packages shown in Figure 4. Each class package corresponds to use case in the Figure 3 use case diagram. The class diagram of "Manufacturing" subsystem describes relationships among the classes with <<usage>> dependency relationship and association relationship. The <<usage>> dependency relationships between class packages show that a change of the class package "ManageItems" may affect to the other class packages: ManageInventories, ManagePurchasing, ManageEquipments, ManageProduction, ManageOutsideProduction, ManageBillOfMaterials, ManageQuality, and ManageProduction, and provide information needed by other class packages [7]. In other words, all the class packages are dependency in the class package of "Manage Items". It indicates that the "ManageItems" class package represents a core class in implementing functionalities of the manufacturing subsystem. And the association relationship represents relationship between classes including in the each package.

Fig. 3 Use case Diagram of "Manufacturing" subsystem

In the Figure 3, the authors describe use case diagram focus on "Manufacturing" subsystem. As mentioned the previous section, ERP is the successor to Material Resource Planning II and had its origins in manufacturing and production planning systems. In addition, the authors developed a conceptual ERP model for small and medium-size manufacturing companies. Therefore, the authors believe that "Manufacturing" subsystem is core of the conceptual ERP model and other subsystems can be expand from this. As looked in the Figure 3, the "Manufacturing" subsystem consists of nine functionalities: ManageItems use case, ManageBillOfMaterials use case, ManageInventories use case, ManagePurchasing use case, ManageProductPlans use case, ManageProduction use case, ManageOutsideProduction use case, ManageEquipments use case, and ManageQuality usecase.

In the use case diagram, relationships between use cases are represented by using stereotype <<include>> and <<extend>>. For example, the ManageBillOfMaterials use case and the ManagePurchasing use case share functionality of the ManageItems use case. When needed the data of outside production, the ManageProduction use case can be extended to the ManageOutsideProduction use case. And Actors, TradingSubsystem, SalesSubsystem, AccountingSubsystem and HumanResource and PayrollSubsystem, represent the external systems related with "Manufacturing" subsystem. In other words, each subsystem can be an actor related with other subsystem.
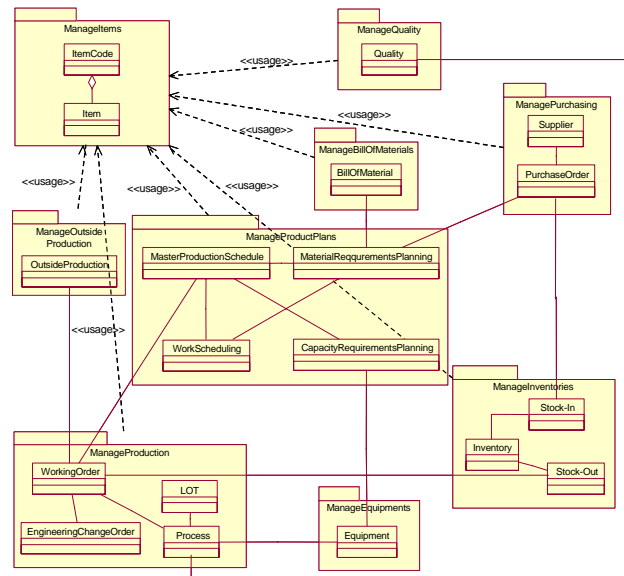

Fig. 4 Class Diagram of "Manufacturing" Subsystem

Classes in each class package aren't described in detail: any attributes and operations are not included. Because of elaborate description of classes is not important in conceptual modeling. It is important in the designing and implementation phase.

4.3 Process View

A sequence diagram illustrates how objects interact with each other. It emphasizes on how messages are sent and received between objects [8][17]. To represent the example of sequence diagram, we choose "ManageInventories" among use cases of "Manufacturing" subsystem show.
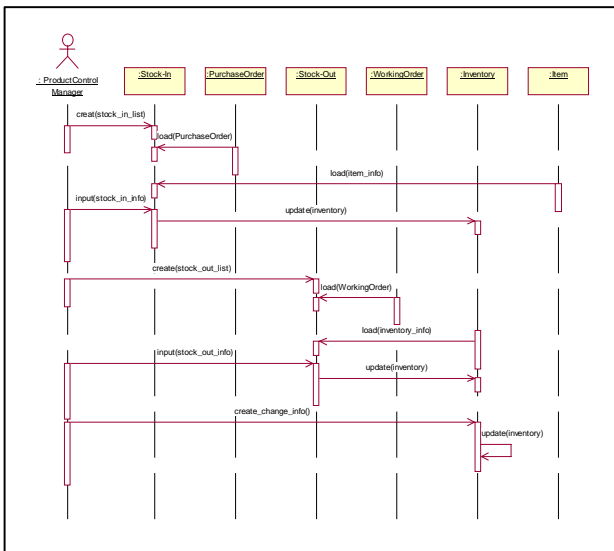

Fig. 5 Sequence Diagram of "Inventory" Use Case

Many scenarios can be described in a single use case and one scenario is related to one sequence diagram. So a use case can have many sequence diagrams. In the Figure 5, the sequence diagram simply represents primary scenario to related with inventory management except for alternative, error, and extension. For example, the created stock-in list is received data from :PurchasOder and :Item object. And the information of :Stock-In upgrades :Inventory class. This is a process of "Manage Stock-In" function in the "Manage Inventories" use case.

4.4 Deployment View

A deployment diagram shows the physical description of the system topology, including the structure of the hardware units and software that executes on each unit [8][17]. We select 2-tiered Client/Server architecture for ERP system's hardware structure. Because small and medium-size companies have a weak in the budget, they want cheap ERP system.

To adopt the patterns that describe the relationships between subsystems, we choose the pipe and filter architectural pattern shown in Figure 6. In the pipe and filter architectural pattern, each piece is independent on each other and dependent on the only date. Therefore without changing the related subsystem, subsystems can be added and replaced.
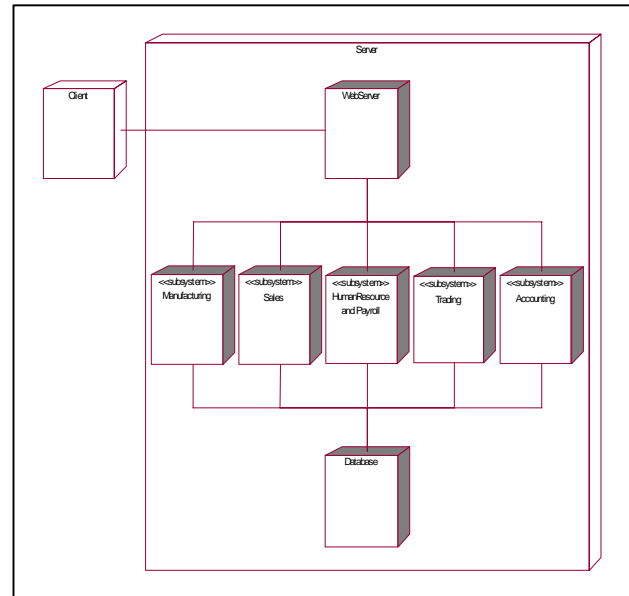

Fig. 6. Deployment Diagram of the conceptual ERP Model

As looked in the Figure 6, it includes two nodes: Client and Server. The server node consists of Web, Database and five subsystems. The Database manages integrated data related with all subsystems and Web has a role to connect Client and subsystems. And adopting the Web technology to ERP is important since internet business is issued in the Information Technology and business area.

In the viewpoint of customers, customer can easily access to the ERP system and the web application server gives business application developers the flexibility to combine ERP functionality with the other data sources and to inject new business logic into an application without changing anything in the ERP processes [16]. Small and medium-size company is not plenty of budgets. They want to reduce hardware cost for adopting ERP systems. So server consists of one hardware piece including web, application, and database server. For example, MS Access is used for database server.

4.5 Implementation View

A component diagram shows software components and their dependencies to each other, representing the structure of the code. The components are the implementation in the

physical architecture of the concepts and the functionality [8]. The component diagram is not founded in analysis phase. Since conceptual model is commonly results from the analysis phase. In other words, the conceptual model has the advantage of strongly emphasizing a focus on domain concepts, not software entities, such as component of file type. Component diagram can be captured after implementation phase. Until now, we show several diagrams and explain each diagram briefly. The reason we use the "4+1" views for describing conceptual ERP model is that views of stakeholders are different according to perspectives of looking at ERP system. In other words, the "4+1" views of architecture are the best way to represent viewpoints of stakeholders.

5. Findings and Implication

ERP system integrates customer relations, finance, manufacturing, inventory, sales, human resources and payroll, field service and any other business areas: "getting all the systems to talk to each other," explained Sean Fleming [1]. Also it provides data integrity throughout the business processes. Now, ERP system is the most common application for all industries, especially manufacturing company to adopt changed business environments. In this paper, the authors represent a conceptual ERP model by using UML as an Object-Oriented modeling technique. By using the conceptual ERP model, ERP developers for small and medium-size companies can obtain many advantages.

The conceptual ERP model along with design and implementation can be used during the whole software life-cycle because the boundaries between analysis, design and implementation are not rigid [10]. It is called seamlessness. The seamlessness should give considerable benefits: flexibility and traceability make ERP systems better quality. It also is much easier to maintain ERP systems because a requirement change can be traced easily. And the artifacts of initial phase would be used during the software life-cycle without additional reworks on results and then software developing process, especially implementation process, is enhanced and become more efficient [15]. In addition, due to reuse of the artifacts of previous phases, software development process has been improved through the elimination of some steps of software developing [19]. The conceptual ERP model simplifies the functionality of ERP systems. The simplification of system makes ERP requirements easier to understand. Commonly, a better system is easier to understand, implement and maintain for the users and the developers.

The authors choose pipe and filter architectural pattern, so the functionality newly needed in the business can be inserted in the ERP model more easily as requirement of business changes [11]. In other words, it is possible to add and delete subsystems at any time without having to change the other subsystems. Each subsystem, described by use case in the use case diagram at high level, is independent on each other. Small and medium-size companies are inferior in organization, human resource, and using the international business process to large companies. The conceptual ERP model could help for developers of small and medium-size companies to understand ERP systems clearly and develop ERP system more easily. Adopting the ERP model, small and medium-size companies can acquire ERP systems more effectively.

References

[1] C. Willard, "ERP's Promised Lands," Computerworld, Oct, 1999

[2] C.P. Holland and B. Light, "Global Enterprise Resource Planning Implementation," Proceedings of the 32nd Hawaii International Conference on System Sciences, pp. 1-10, 1999

[3] C.P. Holland and B. Light, "A Critical Success Factors Model For ERP Implementation," IEEE Software, pp. 30-36, May/June 1999

[4] C. Casanave, "Business-Object Architectures and Standards,"http://dbdoc.ajou.ac.kr/cetus/oo_business_objects.html

[5] G. Bylinsky, "The challengers move in on ERP," Fortune, pp. 306c-313c, Nov 1999

[6] G.S. Lawson, "Enterprise Resource Planning(ERP)-Breakthrough or Buzzword?," Oracle, p291-295

[7] I. Jacobson, G. Booch, and J. Rumbaugh, The Unified Modeling Language Reference Manual, Addison Wesley, 1999

[8] I. Jacobson, G. Booch, and J. Rumbaugh, The Unified Modeling Language User Guide, Addison Wesley, 1999

[9] I. Jacobson, G. Booch, and J. Rumbaugh, The Unified Software Development Process, Addison Wesley, 1999

[10] J.M. Jezequel, Al.L. Guennec, and F. Pennaneac'h, "Validating Dsistributed Software Modeled with the Unified Modeling Language," The Unified Modeling Language: first international workshop selected paper/UML '98 : Beyond the Notation, Springer-Verlag, pp.365-377, 1999

[11] J. W. Ross, "Surprising Facts About Implementing ERP," IT Pro, IEEE, pp. 65-68, 1999

[12] J. Sutherland, "Business Objects in Corporate Information Systems," ACM Computing Surveys, Vol 27, No 2, June, 1995

[13] J. Arlow, W. Emmerich, and J. Quinn, "Literate Modeling-Capturing Business Knowledge with the UML," The Unified Modeling Language: first international

workshop selected paper/UML '98 : Beyond the Notation, Springer-Verlag, pp.189-199, 1999

[14] Joshua, "The Origin and Future of ERP Outsourcing," http://www.erp-outsourcing.com

[15] M. Hitz and G. Kappel, "Developing with UML-Some Pitfalls and Workarounds," The Unified Modeling Language: first international workshop selected paper/UML '98 : Beyond the Notation, Springer-Verlag, pp.9-20, 1999

[16] M. Marshall, "Web application servers give green light to ERP," Informationweek, CMP Media Inc., Apr, 1999

[17] M.M. Kande, S. Mazaher, O. Prnjat, L. Sacks, and M. Wittig, "Applying UML to Design an Inter-domain Service Management Application," The Unified Modeling Language: first international workshop selected paper/UML '98 : Beyond the Notation, Springer-Verlag, pp.200-214, 1999

[18] P.l Hruby, "Structuring Specification of Business Systems with UML (with an Emphasis on Workflow Management Systems," OOPSLA '98 Business Object Workshop
IV,http://jeffsutherland.org/oopsla98/pavel.html

[19] P. Desfray, "Automation of Design Pattern:Concepts, Tools and Practices," The Unified Modeling Language: first international workshop selected paper/UML '98 : Beyond the Notation, Springer-Verlag, pp.120-131, 1999

[20] P. Kruchten, "Architectural Buleprints - The '4+1' View Model of Software Architecture," http://www.rational.com/uml/resources/whitepapers

[21] R. Youngs, D. Redmond-Pyle, P. Spaas, and E. Kahan, "A standard for architecture description," IBM SYSTEMS JOURNAL, Vol 38, No 1, pp.32-50, 1999

[22] R.E. Chalmers, "Small manufacturers seek best ERP fit," Manufacturing Engineering, Dearborn, pp. 42-46, Oct 1999

[23] R. Malan and D. Bredemeyer, "Functional Requirements and Use case," 1999

[24] S. Buckhout, E. Frey, and J. Nemec Jr, "Making ERP Succeed: Turning Fear into Promise," IEEE Engineering Management Review, Strategy and Business, Second Quarter, pp.60-72, 1999

[25] T.F. Gattiker and D.L. Goodhue, "Understanding the Plant Level Costs and Benefits of ERP: Will the Ugly Duckling Always Turn Into a Swan?," Proceedings of the 33nd Hawaii International Conference on System Sciences, pp. 1-10, 2000

[26] "Mid-sized Firms Face Rough Road with ERP Adoption," The Manufacturing Report, Lionheart Publishing, Inc., May, 1999

[27] "Survey of Manufacturing : Meet the Global Factory", IEEE Engineering Management Review Spring 1999, http://www.economist.com/

Jae-Won Park graduated from Sang-ji University, Korea in 2000. Until 2004, and graduated from graduate school of Soongsil University , Seoul, korea in 2004, Until 2004. Until 2006. he is currently pursuing the Ph.D. degree at Soongsil University where his research focuses on Enterprise Resource Planning, Workflow, Electronic Commerce, Software Process, Software Testing and Software Methodology.

Professor, Nam-Yong Lee, Ph.D. received the doctorate in business administration from *Mississippi State University*, MS, in 1993, with a concentration in software reuse. In 1987, he got a certificate of national professional engineers of IT in Korea. He had served as the director of *Information Systems Directorate at Korea Institute for Defense Information Systems (KIDIS)* and had been in charge of numerous projects associated with National Defense Information Systems Acquisition Projects in Korea and the U.S DoD for about 20 years. Now, he is a professor of *School of Computing, College of Information Science, Soongsil University*. The school is one of the most prestigious and famous schools on IT in Korea. He also is a chairperson of the informatization steering committee of *Information & Communication Professional Engineers Association of Korea(ICPEAK)*. He currently serves as chief-in-editor of the Journal of *Korean Institute of CALS/EC (KICE)*, serves on the editorial boards of *Korea Information Processing Society(KIPS), Korea Information Society(KIS), and Korean Industry Information Society(KIIS),* respectively. He has published numerous books, including *Applications of Ada Programming Language*, Jung-ik Publishing, Inc., *CALS and Electronic Commerce*, Bub-young Publishing Co., *Instant CORBA*, Youngjin Publishing Co., etc. His more than 50 publications have appeared on numerous conference proceedings and academic journals. For example, his publication has appeared on *IEEE Transactions on Software Engineering* entitled in "*A Study on Software Reuse with Special attention to Ada*", on September 1997. His research focuses on software reuse, software metrics, software engineering, electronic commerce, project management based on the object-oriented technologies.