

PROCESS-DRIVEN SOFTWARE DEVELOPMENT METHODOLOGY FOR ENTERPRISE INFORMATION SYSTEM

Kwan Hee Han¹ and Yongsun Choi²

¹Department of Industrial & Systems Engineering,
Engineering Research Institute,
Gyeongsang National University, Jinju, Korea
hankh@gnu.ac.kr

²Department of System Management & Engineering, Inje University, Korea
yschoi@inje.ac.kr

ABSTRACT

In today's process-centered business organization, it is imperative that enterprise information system must be converted from task-centered to process-centered system. Traditional software development methodology is function-oriented, in which each function manages its own data and it results in redundancy because data that belongs to one object are stored by several functions. Proposed in this paper is a process-driven software development methodology, in which business process is a major concern and workflow functionalities are identified and specified throughout the entire development life cycle. In the proposed methodology, the development process, modeling tools and deliverables are clarified explicitly. Proposed methodology can be a guideline to practitioners involved in enterprise software development, of which workflow is an essential part.

KEYWORDS

Business Process, Software Engineering, Enterprise Information System, Object-Oriented, System Development Methodology

1. INTRODUCTION

Nowadays the unpredictability of market changes, the growing product complexity and continuous pressure on costs force enterprises to develop the ability to respond and adapt to change quickly and effectively. To cope with these challenges, most enterprises are struggling to change their existing business processes into agile, product- and customer-oriented structures to survive in the competitive and global business environment. In today's dynamic business environment, the ability to improve business performance is a quintessential requirement for all enterprises [1].

Therefore, many enterprises have recently been adopting enterprise information systems such as ERP (Enterprise Resource Planning), WFM (Workflow Management) and PLM (Product Life cycle Management) system to attain their performance goals.

As business environment is being changed, substantial change is being also occurred in the development of enterprise information system. This is mainly due to the change of management structure as well as change of software development methodology. First of all, many enterprises recognized that it is difficult to survive by Taylor's 'scientific management' under the rapid change of business environment. Therefore, to cope with these challenges, the concept of business process management (BPM) has been recently proposed. BPM is the identification, understanding, and management of business processes linked with people and systems and across organizations [2]. It is an approach to transform traditional functional organization to market-oriented process organization. So, it is needed that enterprise information system must be converted from task-centered to process-centered system. In other words, it is necessary to transfer from existing function-oriented information system supporting individual task to workflow management system integrating entire business process for specific customer satisfaction [3]. Therefore, workflow automation is emerging as an essential part of information system beyond supporting transactional task in the recent enterprise information system development.

With respect to software development methodology, 'software crisis' was occurred due to the ever-increasing complexity of software structure and rapid increase of software development and operational cost. To overcome this crisis, in the area of system development methodology, object-oriented development methodology is widely used as an alternative of existing structured system development methodology that revealed difficulties of systematic integration of each phase's deliverables and efficient development phase transition. Another limitation of existing software development methodology is that there is no common modeling method for developer and end user.

In summary, to align with current process-centered approach of enterprises, integrated software development methodology is needed to deal with workflow automation for efficient business process management and object-oriented modeling for systematic software design simultaneously.

The objective of this paper is to propose a process-driven object-oriented software development methodology by integrating business process modeling and information system modeling which were separately modeled in the conventional development of enterprise information system. Within proposed methodology, it is suggested that UML (unified modeling Language) is suitable to the business process modeling as well as software modeling.

The rest of the paper is organized as follows: Section 2 reviews current practices and limitations of typical software development methodologies. Section 3 describes a proposed process-driven software development framework. Finally, the last section summarizes results and suggests directions for future research.

2. LIMITATIONS OF CURRENT SOFTWARE DEVELOPMENT METHODOLOGIES

Typical software development process for enterprise information system is as follows (Figure 1): (1) Systems analysis, (2) Systems design and (3) Systems implementation and operation.

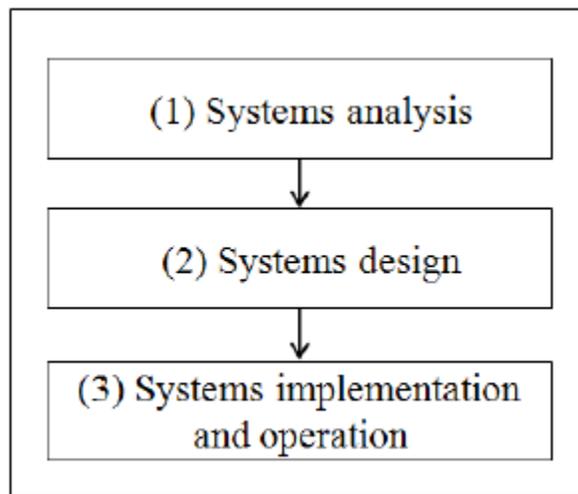


Figure 1. Conventional software development process

Business process is a collection of interrelated work tasks, initiated in response to an event, which achieves a specific result for the customer of the process [4]. Enterprise system consists of organizations performing these business activities.

Therefore, in order to develop enterprise information system, business process analysis is a prerequisite task. Current object-oriented software development process often omits this requirement or deals with this task in the system design phase (Phase-2) slightly.

The task of business process analysis is usually performed by business analysts under the name of business process modeling or workflow modeling during business process reengineering projects. Major modeling tools include IDEF0 [5], EPC [6] and Petri net [3], [7]. As described above, in spite of close relationship between business process modeling and software system modeling, these two tasks are typically performed separately in recent software development practices.

UML (Unified Modeling Language) is a de facto standard modeling tool for object-oriented software development, which was announced by OMG (Object management Group) in November, 1997. The current version of UML is 2.3 (May 2010). It was being rapidly adopted because of its graphical notation, which is readily understood, and a rich set of semantics for capturing key features of object-oriented systems.

Moreover, it was addressed that UML is also appropriate for business process modeling as well as information system modeling by virtue of its expressiveness, user-friendliness, and integration

capability with information systems [8], [9]. Recently, there are considerable active approaches to UML-based business process modeling [8], [9], [10].

However, UML did not define development process since it is only a modeling tool intended for the purpose of software development. As an object-oriented development using UML, Unified process (UP) [11], [12] is widely adopted beyond OMT [13], Booch [14], Ericsson and objectory methodologies. The characteristics of UP are as follows: 1) it is a use case-centered process. 2) It is an architecture-centered evolutionary incremental approach. However, there are not a few substantial difficulties in applying UP to real software development projects in spite of its strengths such as software productivity increase, reusability increase and natural mapping to real world.

3. PROCESS-DRIVEN SOFTWARE DEVELOPMENT METHODOLOGY

Proposed process-driven software development methodology is based on the facts found in current development practices as follows: 1) it is difficult to transfer the deliverables of business process modeling to the process of enterprise information system development. 2) Though UML is used as a de facto standard modeling tool, there is no explicit guidelines about which UML diagram must be used appropriately in the specific development phase.

Since software development methodology is a systematic approach to successfully develop software system, so it must provide process, notation, and tools for software development in a consistent and integrated way. The process of proposed development methodology consists of 6 phases as follows (Figure 2):

- (1) Business process analysis
- (2) User Requirements analysis
- (3) Structure Design
- (4) Architecture Design
- (5) Detail Design
- (6) Implementation and Operation

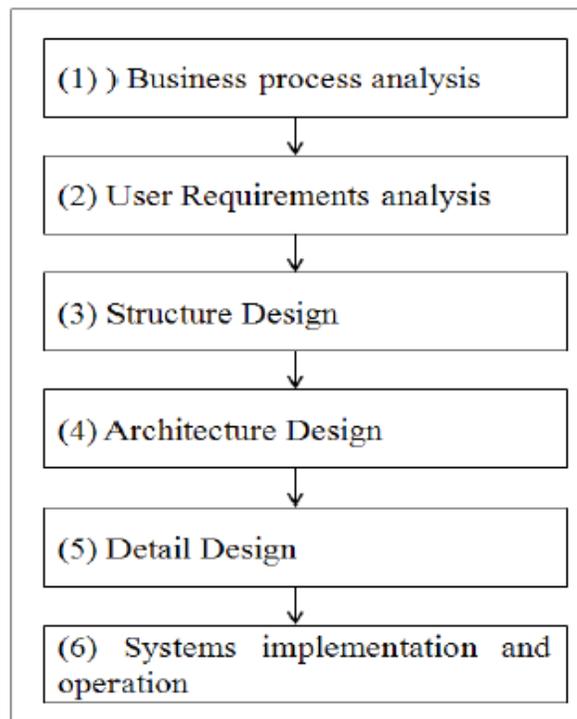


Figure 2. Process-driven software development process

Following subsections describe the framework of each development phase. This framework consists of 3 elements as follows: 1) development process, 2) modeling tool, 3) deliverables (called ‘model’)

3.1. Business Process Analysis Phase

Proposed methodology starts with business process analysis. The main purpose of this phase is to better understand the key mechanisms of existing business and to show the structure of an innovated business.

A business is a complex system, consisting of a hierarchical organization of departments and their functions. Thus, business process analysis focuses on the core business tasks and its key mechanisms. The central concept used for business process analysis is the business process, which describes activities within the business and how they relate to and interact with the resources in the business to achieve a goal for the process.

Before designing new processes, it should be clear which performance aspects of the business process are targeted for improvement. In most circumstances, it would be ideal if a redesign of a business process decreases the time required to handle customer needs, decreases the required cost of executing the business process, improves the quality of service delivered, and improve the ability to change the business process to react to variation.

The UML diagram used in this phase is an activity diagram. Among UML diagrams, the activity diagram is most suitable for business process modeling because of the following features: 1) it

can describe more easily dynamic behaviors of business process and events triggering the process. 2) It can represent various workflow routing types such as sequential routing, join, split, iteration, and parallel routing suggested by WfMC (Workflow Management Coalition).

Deliverables in this phase are ‘as-is’ process and ‘to-be’ process model. As-is process model describes current aspects of business processes, which is a target of information system, whereas to-be process model describes desirable innovative process after installing new enterprise information system. Information systems are key enablers to achieving process innovation or BPR (Business Process Reengineering).

In the activity diagram for describing business process, business object, which is an input and output of certain activity, must be specified explicitly. Business objects are categorized and specified as physical, document, drawing, data set objects by using UML stereotype notation. A swim-lane is divided by actor or organizational unit. During to-be process analysis, more attention is needed to the cross-swim-lane activity flow and the potential parallel processing activities.

3.2. User Requirements Analysis

The main purpose of this phase is to describe what a new system should do or what an existing system already does. A ‘functional model’, which is a deliverable, is built through an iterative process mainly by discussion between the system developers and end users. The actual work required to create a functional model involves defining the system, finding actors and the use cases, describing the use cases, defining the relationships between use cases, and finally validating the model. The UML use case diagram and use case description is used in this phase.

In this phase, user requirements are elicited from to-be process model, and must be converted to functional requirements of information system. To do this, business activities within a process requiring the service of information system should be identified from the to-be process model. Typical activities requiring information system service are as follows: 1) information storage, retrieval, and organizing, 2) information processing, transformation and delivery, 3) decision making, 4) communication, and 5) hardware control.

In other words, use case defines service specifications, which is provided by enterprise information system to business processes.

There are informational requirements for each process, which describes the “what” factor. It creates or modifies objects. Objects represent things that are described both by properties and by the method that can be applied to them. They are specified at the structure design phase in more detail.

Functional model describes what is required of a system by defining how it will be used by external actors. An actor, which is usually but not necessarily human, causes the system to perform its functions by means of use cases.

A business use case may be thought of as a collection of related process steps, and actors may be thought of as the organization roles that execute the process steps. Because a business process

defines how a purpose is to be achieved, a business use case should also be directed to satisfy a business purpose.

Most modern enterprise information system requires workflow functionalities since the workflow loop is the basis of all business processes. Workflow is defined as the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules [15].

Therefore, in this phase, workflow type must be determined for the business process automation. There are two cases when developing enterprise information system as follows: 1) concurrent development of new transaction processing system and workflow system, 2) additional development of workflow system on the existing information system.

With respect to the degree of coupling between systems, there are two types as follows [16]: 1) autonomous workflow management system, 2) embedded workflow management system. Autonomous system, which is independent from other information system, executes workflows by invoking application software when needed. Embedded system can only be operated within its environmental system such as ERP and PDM (Product Data Management) system.

Though the workflow type selection is dependent to enterprise operating environment, autonomous workflow management system is advantageous when there exist several heterogeneous information system within enterprise.

3.3. Structure Design

The main purpose of this phase is to describe the internal structure of information system satisfying user requirements in terms of domain concepts called class. By doing this, we can understand the features of generated data from new information system.

The UML class diagram is used in this phase, and deliverable is a 'structure model'. In this phase, major task is to identify responsible classes for functionalities of each use case described in the functional model, and to establish a relationship between classes undertaking same responsibilities. Once basic classes responsible for use case are identified, then associated classes are investigated sequentially. Major candidates of basic class are business objects identified in the business process analysis phase.

In general, class is categorized into 3 types such as entity, boundary and control class [12]. In this paper, control class is further extended to 2 subtypes of logic control class and workflow class. Former class deals with business logic, and the latter deals with the flow of business process.

Boundary class is in charge of interactions between information system and external actor. According to the characteristics of information processing requirements, boundary class is further classified into data-centric and document-centric class. If it is data-centric, boundary class is converted into general input/output screen in the detail design phase. If it is document-centric, it is designed as electronic form.

Entity class deals with static and persistent information required by enterprise when performing business activities.

Control class is in charge of executing business rules and sequencing and coordinating business activities. Logic control sub-class has a responsibility of detecting exceptional situation, checking the condition of business rules and settling the attribute value and so on. Workflow control class performs sequencing and coordination of business activities. In other words, it enables and control access to knowledge and information while coordinating all the roles in a business process.

The establishment of structure model having well-defined workflow functionalities is very essential in today's modern process-oriented organization. Workflow is a key enabler of BPR or PI (Process Innovation). The major benefits of workflow management are as follows:

- 1) Improved efficiency: automation of many business processes results in the elimination of many unnecessary steps.
- 2) Better process control: improved management of business processes is achieved through standardizing working methods and the availability of audit trails.
- 3) Improved customer service: consistency in the processes leads to greater predictability in levels of response to customers.
- 4) Flexibility: software control over processes enables their redesign in line with changing business needs.
- 5) Business process improvement: focus on business processes leads to their streamlining and simplification.

3.4 Architecture Design

Information system architecture is a description of entire resources including hardware, software and network, and their relationships for fulfilling organization's information needs. In particular, the term software architecture intuitively denotes the high level structures of a software system.

It can be defined as the set of structures needed to reason about the software system, which comprise the software elements, the relations between them, and the properties of both elements and relations [17].

The main purpose of this phase is to describe the basic structure of software system and relationship between software elements, and to suggest guidelines to the detail design phase.

UML class diagram and UML deployment diagram are used in this phase, and deliverable is an 'architecture model'. Software architecture is represented by UML class diagram, and hardware/network architecture is represented by UML deployment diagram. In the modern client-server computing, overall software structure is mainly designed as a hierarchical structure [18]. Currently, 3-tier architecture is widely used in developing enterprise information system [19].

In the layered pattern, sub-system is represented by hierarchical layer responsible for system functionalities as follows: 1) interface layer is for user interface, 2) application layer for business logic, 3) storage layer for persistent data management.

Recently, interface layer is subdivided to presentation layer and interface layer to reflect various presentation media such as web browser and smart phone. Storage layer can be extended further to service layer for common functionalities such as message handling, security check, and access control.

3.5. Detail Design

The main purposes of this phase are as follows:

- 1) To describe how internal entities interact with each other, and when their state is changed.
- 2) To refine structure model by specifying operations and attributes of classes.

UML sequence diagram, communication diagram, state diagram and class diagram are used in this phase, and deliverables are 'dynamic behavior model' and 'detail design structure model'. UML sequence diagram or communication diagram is used to represent how internal entities exchange messages with each other when performing their duties. UML state diagram is used to represent state change of internal entities.

Whereas functional model focuses on functional view, and structure model focuses on internal structural view, detail design model focuses on dynamic view according to passage of time.

In this phase, main tasks are to identify the specific operations of each domain entity and their state change. During defining operations for undertaking responsibilities of domain entity in response to external events, if temporal sequence of each operation is important, UML sequence diagram is used.

Otherwise, if collaboration relationship between domain entities is important, UML communication diagram is suitable. For identifying state changes of domain entity, UML state diagram is used.

After specifying operations of domain objects in the dynamic behavior model, 'structure model' is refined to 'detail design structure model' on which operations of each entity is added and specified.

Besides, a class identified as boundary class in the structure model is also refined to reflect user interface design. It is designed as input/output screen or electronic form. If persistent storage is required, additional classes for the database implementation are also added to detail design structure model.

In summary, Figure 3 shows the development process, UML diagrams and deliverables of proposed process-driven software development methodology.

Development Process	UML Modeling Tool	Deliverables
(1) Business process analysis	-activity diagram	-‘as-is’ process model - ‘to-be’ process model.
(2) User Requirements analysis	-use case diagram -use case description	-functional model
(3) Structure Design	-class diagram	-structure model
(4) Architecture Design	-class diagram -deployment diagram	-architecture model
(5) Detail Design	-sequence diagram, -communication diagram -state diagram -class diagram	-dynamic behavior model -detail design structure model

Figure 3. Tool & deliverables of process-driven software development methodology

3. CONCLUSIONS

External environment of enterprise are rapidly changing brought about majorly by global competition, cost and profitability pressures, and emerging new technology. Current environment and their impact on the organizations are that product’s lifetime is shorter, product has customer specific versions, open market requires cost/lead-time reduction, and organizations become more internationalized, think globally. To cope with these challenges, most enterprises are transforming their organization from functional and hierarchical one to process-centered one for customer satisfaction. To support process-centered organization, enterprise information system must be developed in process-driven way.

Traditional software development methodology is function-oriented, in which each function manages its own data and it results in redundancy because data that belongs to one object are stored by several functions. Additionally, workflow automation is emerging as an essential part of information system beyond supporting transactional task in the recent enterprise information system development.

Proposed in this paper is a process-driven object-oriented software development methodology, in which business process is a major concern, and workflow functionalities are identified and designed throughout the entire development life cycle. In the proposed methodology, the development process, modeling tools and deliverables are clarified explicitly.

Therefore, proposed methodology can be a guideline to practitioners involved in enterprise software development, of which workflow is an essential part.

However, Usefulness of proposed methodology must be validated in real development projects, which is one of further research in near future.

ACKNOWLEDGEMENTS

This work was supported by the Nuclear Power Core Technology Development Program of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) granted financial resource from the Ministry of Trade, Industry & Energy (MOTIE), Republic of Korea (No. 20131510101690: Development of a public monitoring system and competence enhancement technologies for strengthen Korean safety culture in nuclear organizations).

REFERENCES

- [1] Han K. H, Kang J. G. and Song M. (2009), "Two-stage process analysis using the process-based performance measurement framework and business process simulation," *Expert Systems with Applications*, vol. 36, no. 3, part 2, pp. 7080-7086.
- [2] Debevoise T. (2005), *Business process management with a business rules approach*, Roanoke, VA, USA: Business Knowledge Architects.
- [3] Aalst W. M. P. (1998), "The application of Petri nets to workflow management," *The Journal of Circuits, Systems and Computers*, vol. 8, no.1, pp. 21-66, 1998.
- [4] Sharp A., McDermott P. (2001), *Workflow Modeling*, Boston: Artech House.
- [5] National Institute of Standards and Technology (1993), *Integration definition for function modeling-draft federal information processing standards publication 183*, NIST, USA.
- [6] Scheer A. W. (1998), *Business process engineering*, Berlin, Germany: Springer-Verlag.
- [7] Han K. H., You S. K. and Kim B. H. (2009), "Qualitative and Quantitative Analysis of Workflows Based on the UML Activity Diagram and Petri Net," *WSEAS Transactions on Information Science and Applications*, vol. 6, no. 7, pp. 1249-1258.
- [8] Eriksson H. E., Penker M (2000), *Business modeling with UML*, New York: John Wiley & Sons, Inc.
- [9] Marshall C. (2000), *Enterprise modeling with UML*, Reading, MA: Addison Wesley.
- [10] Hruby P. (1998), "Structuring specification of business system with UML," *OOPSLA'98 Business Object Workshop IV*, Vancouver, British Columbia, Canada.
- [11] Kruchten P. (1998), *The rational unified process: an introduction*, Reading, MA: Addison-Wesley.
- [12] Jacobson I., Booch G. and Rumbaugh J. (1999), *The unified software development process*, Reading, MA: Addison-Wesley.
- [13] Rumbaugh J. et al (1991)., *Object-oriented modeling and design*, Englewood Cliffs, New Jersey: Prentice Hall..
- [14] Booch G. (1994), *Object-oriented design with application*, Benjamin/Cummings. Redwood city, California:
- [15] WfMC (1999), *Workflow Management Coalition Terminology & Glossary*, Document Number WfMC-TC-1011, Workflow management Coalition, 1999.
- [16] Muehlen, R. A. (2000), "Workflow classification: embedded & autonomous workflow management systems," *Workflow Management Coalition*.
- [17] Clements P., Bachmann F., Bass L., Garlan D., Ivers J., Little R., Merson P., Nord R. and Stafford J. (2010), *Documenting Software Architectures: Views and Beyond*, Second Edition. Boston: Addison-Wesley.
- [18] Larman C. (2005), *Applying UML and patterns*, Third Edition, New Jersey: Prentice Hall PTR.
- [19] Shuttle (1995), *Three tier computing architectures and beyond*, Gartner Group.

AUTHORS

Kwan Hee Han is a professor in the department of industrial & systems engineering, engineering research institute of Gyeongsang National University in Korea. His research interests focus on simulation modeling, virtual manufacturing, business process management and process mining, etc.



Yongsun Choi is a professor in the department of system management & engineering of Inje University in Korea. His research interests include process modelling and analysis, decision support systems, internet of things, multiple criteria optimization, etc.

