Enterprise Process Reverse Engineering (EPRE)
: Form-Based Approach

Kyong-Ho Kim, Young-Gul Kim
Graduate School of Management
Korea Advanced Institute of Science and Technology
Seoul, Korea
hikhk@unitel.co.kr, domino2@unitel.co.kr

Abstract

Firms spend enormous efforts identifying current processes and understanding the related details before establishing new business processes in their business process redesign (BPR) projects. Considering excessive efforts required during the analysis phase and limited support to BPR projects, need for a better method is evident. This article describes a method of modeling enterprise business processes based on common business forms. By identifying form operations and form field types, the proposed method provides redesign view on the information handling as well as the current process flows in the form of an Event-Process Chain (EPC) diagram.

1. Introduction

When we perform business process redesign (BPR) projects, we have to analyze the current process flows to identify the potential problem areas before creating a new process or redesigning the existing ones. However, it is not a trivial task to identify and represent those processes in a formal, yet easy to understand, process model. Moreover, capturing and fully modeling the business processes usually depends on the discovery approach such as interviewing people, which is absolutely time consuming and leads to the inaccuracy and inconsistency[4]. Furthermore, we suffer from the limitation in the support of the modeling formalism for the process redesign. Because most process formalisms originated from the organizational or the application development perspective, they could not satisfy the cross functional, customer-oriented process nature of BPR project. Considering these excessive efforts required during the analysis phase of BPR projects, need for a better method for the current process modeling and redesign support is evident. This paper introduces a method that helps analyze target business processes in the BPR perspective. The proposed method will greatly facilitate the interaction between systems analysts and end-users during the "modeling" phase of the BPR implementation process.

The rest of the paper proceeds as follows : section 2 discusses the previous research. In section 3, we describe the enterprise process model that will represent the existing process flows. Section 4 describes the detail stages and steps that lead to the expected process flows, together with a sample case. Section 5 summarizes the proposed method and suggests future research directions.

2. Previous research

The basis of this paper originates from the combination of process modeling formalism and form-based reverse engineering concepts. In the research regarding to the form-based reverse engineering, process focused research are earlier than others such as data model focused research, since form routing naturally shows process flows with organizational behaviors[9]. However, their main motivation is how to manage forms in the server organization perspective. Therefore most process focused research tried to get the form routing and to implement automatic form procedures[1][9]. In the dynamic modeling research field, the document is also the basic resource in identifying interactions between organizations. The documentary procedures are redefined as commitment procedures involve communication by means of structured documents[8]. The workflow management researchers also have interests in the process flow that can be inferred from the form's information[4].

3. Form definition and enterprise process model

3.1 Scope and form definition

As for the activities performed by the organization, we can categorize them into three types : information handling activity, physical and information handling activity, and physical activity. We will confine the scope of the proposed method to the first two types of activities of which information handling is a required component, because use of forms inherently requires the target tasks to handle information. Choobineh[2] defines a form as "a structured collection of variables (i.e., form fields) that are appropriately formatted for data entry and display". We would use Choobineh's definition, because it is very logical and can be applied to both electronic and other traditional paper forms. Accordingly, our form-based process modeling method is applicable to all kinds of forms used in business.

3.2 Enterprise process model : Event-Process Chain

When the business process redesign (BPR) paradigm was introduced in early 1990s, process modeling formalism found a new role : from a model for function-specific automation to a cross-functional model for radical process redesign from the customer's view point. However, most of them could not satisfy the cross-functional, customer-oriented process nature of BPR projects effectively. The Event-Process Chain (EPC) modeling formalism we adopted in this study was exclusively designed to support BPR and it is the only process modeling formalism for BPR where the customer's perspective is fully supported[5].

4. Process modeling procedure

The process modeling procedure consists of the following three stages as in Table 1. We will use a hospital case to explain the concepts adopted by our methodology .

Table 1. Methodology step summary

Stage	Step
1. Field type identification	1. Identify user information 2. Identify field set operations 3. Identify form field types
2. Operation flow generation	1. Generate operation names 2. Generate operation flow 3. Identify processing/waiting time
3. EPC generation	1. Generate EPC and specification 2. Complete EPC

4.1 Stage 1 - Field type identification

Step 1 : Identify user information. The first step of the first stage is identifying user information such as user department and working place for the place dimension of the process model, which will satisfyorganizational perspective.

Step 2 : Identify field set operations. To explain actions requested or performed by a customer or by a server, we define a *Field Set Operation (FSO)* as "a set of activities which process a subset of the form field and performed at a single location during a single service session for a specific purpose". For each FSO, the information about whether it deals with a customer directly or not is also required for providing redesign view at the end of the method. As for activities of an FSO, each activity has its own activity type for processed field values: *create, update, query* and *delete.*

Step 3 : Identify form field types. After defining FSOs, the type of each field processed by each FSO should be identified according to the two type dimensions : *Origin Type* and *Input Type*. *Origin Type* explains where the field value originates while *Input Type* shows how the field value is generated as shown in Table 2.

Table 2. Field type definition

Dimension	Category	Definition
Origin Type	*N* (New)	Input field value is not delivered from any existing form instance(s) within the firm
	E (Existing)	Input field value should be delivered from already existing form instance(s) within the firm
Input Type	*U* (User-input)	User inputs the field value
	S (System-provided)	The field value is automatically assigned by the system

We will represent the field type of a certain form field as [value of *Origin Type, value of Input Type*]. The important characteristic of the field type is that the field type of a certain field varies by each FSO. However in another operation, for instance the consultation, This characteristic of the field type provides us with the redesign view on the information handling activities. Those fields categorized as [*Existing, User-input*] generate the possibility of inefficiency and ineffectiveness in information handling activities. As for inefficiency, [*Existing, User-input*] means that users must input field value again even though it exists somewhere in the firm. This absolutely leads to time consuming and causes potential incorrectness. If possible, this type should be avoided by changing them into [*Existing,System-provided*]. About ineffectiveness, we can think about *the opportunity cost* in the customer perspective. Assume that a customer is supposed to input his card information as a payment method for each ordered item, it may persuade customers not to use that form mainly due to inconvenience, which leads to the reduction of customer orders.

4.2 Stage 2 - Process flow generation

Step 1 : Generate field set operation names. After analyzing field set operations, we can generate their names based on related form names. However, we have to select main form to generate a representative name if an FSO is related to more than two forms. The main form will be one of the forms of which fields are processed by 'create' activity.

Step 2 : Generate operation flow. In the second step, operation sequence between two FSOs will be determined by the field type information of a certain field which exists in both FSOs. We will call that field *a common field*. There are 2 basic criteria for determining the operation sequence. The first criterion is based on Origin Type of a common field. Assume that there are two FSOs. The first one is Registration that processes the registration # field classified as *New* type while the second one is Consultation that also processes that field classified as *Existing* type. Then we can say Registration occurs first, and Consultation occurs next, because a value of a common field should be created prior to the execution of Consultation. The second criterion is based on the activity type performed on a common field. That is, 'create' activity should precede to other activity types such as 'update', 'query', and 'delete'. However, the occurrence priority among update, query, and delete activity is not determined automatically.

Based on the operation sequence, we can decide optional sequence: if the value set of a common field in FSO *O1* is not always same to that of a common field in FSO *O2*, then we can say that sequence is branching.

Step 3 : Identify processing time and waiting time. After finishing above 2 sub steps, we are ready to get a Customer-Server Service Flow Diagram(CSSFD) that shows overall service situation toward a customer. Then, the processing time for each process unit and the waiting time for each interval between processes are required [5]. After identifying each time, the model is going to show where the potential problems exist based on the time and information handling usage. The real-line arrow shows the service from server organization to the customer directly, while dotted line explains the server organization deals with customer object that takes the customer's role in dealing with a server.

4. 3 Stage 3 - EPC generation

Step 1 : Generate Event-Process Chain and specifications. We can now generate first-level EPC which is entirely expressed in the customer perspective. Each FSO is converted into a process in EPC with the identified processing time. Certain processes with short processing times can be changed into events automatically. Figure 1 shows the first-level EPC in the hospital case. The nature of aggregated wait is revealed in the form of a lower level EPC diagram only when it is unacceptable to the customer. For instance, if W5 which consists of several waits and processes is unacceptable to the customer, W5 is exploded into a lower level EPC diagram where the patient's medicine request replaces the role of the customer.

Step 2 : Complete Event-Process Chain. The last step of the method is to make the first-cut EPC completed. Since the first-cut EPC does not consider physical activities that do not deal with any information handling activities, and there may be some erroneous result such as a wrong process name, users are supposed to complete EPC by adding physical activities and correcting process names.

5. Summary and conclusion

We proposed a methodology to generate current enterprise process model by analyzing forms through Event-Process Chain which is exclusively

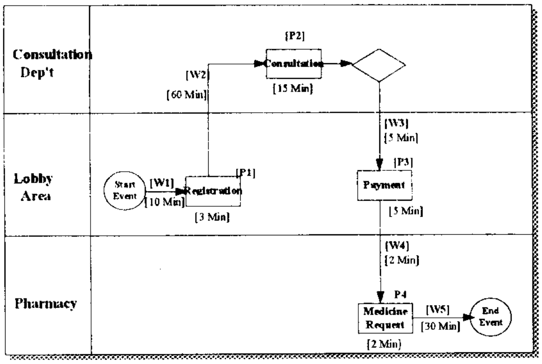

Figure 1. First-cut Event-Process Chain : Level-1

designed to support BPR from the customer's pespective. We regarded a form as a service or the interaction mechanism between a customer and a server. Our model consisted of three stages. 1) Field type identification. 2) Process flow generation. 3) Event-process chain generation. The result of the method satisfied the process modeling components in terms of functional, behavioral, organizational, and informational perspectives [3].

However, since our method excludes physical activities due to the form based approach, the method needs user input for physical activity identification. Also, the method has limitation in extraction of various types of flows such as exclusive OR, because the method tries to extract operation flows based on simplified 2 dimensions of the field type.

The proposed form-based process modeling is still at the conceptual level. Therefore, the remaining task is to build a prototype system for our method and apply it to real world cases. Another potential future direction of this research is to extend our methodology into the areas of the hypertext form and object-oriented model. Since the object-oriented model integrates the data model and process model, and forms also provide both data and process information, we might be able to extend our method so that we can generate object-oriented models based on form analysis.

Reference

[1] Celentano, A., et al. "Knowledge-Based Document Retrieval in Office Environments: The Kabiria System," *ACM Transactions on Database Systems,* Vol. 13, No. 3, Jul. 1995, pp 237-268.

[2] Choobineh, J., Mannino, M., and Tseng, V. "A Form-Based Approach for Database Analysis and Design," *Communication of the ACM,* Vol. 35, No. 2, Feb. 1992, pp 108-120.

[3] Curtis, B., Keliner, M. I., and Over, J. "Process Modeling," *Communications of the ACM,* Vol. 35, No. 9, Sep. 1992, pp 75-90.

[4] Georgakopoulos, D., Hornick, M, and Sheth, A. "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure," *Distributed and Parallel Databases,* Vol. 3, 1995, pp 119-153.

[5] Kim, Y. "Process Modeling for BPR - Event-Process Chain Approach," *Proceedings of the 16th International Conference on Information System,* 1995, pp 109-121.

[6] Kim, H. and Kim, Y. "Dynamic Modeling with Simulation:BPR Application," *Proceedings of the 1st APDSI, 1996,* pp1031-1310.

[7] Kornatzky, Y. and Shoval, P. "Conceptual Design of Object-Oriented Database Schemas using the Binary-Relationship Model," *Data & Knowledge Engineering,* Vol. 14, 1994, pp 265-288.

[8] Lee, R. M. "Dynamic Modeling of Documentary Procedures: A Case for EDI," *Proceeding of the 3rd Internatl Working Con. on Dynamic Modeling of IS,* Jun. 1992, pp 95-123.

[9] Shu, N. C., Lum, V. Y., Tung, F. C., and Chang, C.L. "Specification of Forms Processing and Business Procedures for Office Automation," *IEEE Transactions on Software Engineering,* Vol. 8, No. 5, Sep. 1982, pp 499-512.

[10] Tsichritzis, D. "Form Management," *Communication of the ACM,* Vol. 25, No. 7, Jul. 1982, pp 453-478.