Proceedings of the Thirtieth Annual Hawaii International Conference on System Science, 199-209, (1997)

A Cognitive Engineering Study on the Development of an Object-Oriented Process Modeling Formalism

Hyoungmee Hahn Jungpil Hahn Jinwoo Kim

Cognitive Information Engineering Lab

School of Business Administration

Yonsei University, Seoul, 120 - 749, Korea

hmhahn@infolab.yonsei.ac.kr, jphahn@bubble.yonsei.ac.kr, jinwoo@bubble.yonsei.ac.kr

Abstract

The task of Business Engineering (BE) would be effective if provided with tools that support the analysis and understanding of a business system. This study aims at 1) empirically investigating the cognitive processes of analyzing multiple models of the object-oriented methodology, 2) identifying the problem solving strategy for understanding a business system, and 3) proposing a modeling formalism that fits the problem solving strategy. The results of this study show that existing object-oriented methodologies need to be revised for providing integrated and multi-dimensional views of the core business process. We suggest low-level use case diagrams and time, cost and quality dimensional message trace diagrams to provide a comprehensive modeling formalism of object-oriented business engineering.

1. Introduction

The business environment of today is characterized by rapid flux [1]. Customers have become more sophisticated and demanding, competition has turned global and ferocious. This kind of change has put organizations in a situation where process innovation [2] or business process reengineering [3] is the only key to survival.

However, business process reengineering (BPR) is not an easy ride. Fifty to seventy percent of all reengineering projects fail. There are many reasons as to the high rate of failure such as under-emphasis on processes, lack of executive leadership, and premature implementation of the redesign. Besides these soft factors other factors play an important role in the actual reengineering process. The development of the business system, which is the model of the real world business process, should be closely coupled with the development of the information system that supports the business system. Inability to model concurrently the business system and information system causes needless iteration of analysis and design tasks for each of the systems, and discontinuity between the two development processes which lead to a high rate of failure.

Therefore organizations today need to adopt a business engineering (BE) approach that incorporates the modeling and design of the business system (i.e. the organizational structure and the business processes) and information systems. Business Engineering is an organizational transformation that focuses on the integral design of both information systems and organizational systems [4]. Business engineering can be approached from three different perspectives; macro, meso and micro perspectives [5]. The macro-perspective focuses on engineering the infrastructure between different organizations. The meso-perspective concentrates on engineering processes and coordinating activities within the boundaries of an organization, which is the focus of this study. The micro-perspective focuses on unit tasks that are usually performed by an individual or a small group. From the meso-perspective, organizations need methodologies to guide them through sound engineering and coordinating processes. Notably, methodologies provide formalisms to help model and understand the business system and information system being engineered. The formalism consists of the architecture and the notation as a means of analyzing and designing the systems [6][7]. Organizations need methods that provide such architecture and notation that help model their business processes and also support the cognitive processes of the reengineering team that analyzes and redesigns the business processes [8].

The objective of this study is to develop a process modeling formalism that effectively supports the cognitive processes of the business engineer when understanding business processes for business engineering. The study consists of two phases. First, we have conducted experiments to investigate the effectiveness of the object-oriented methodology for business engineering. We sought to uncover through our experiment the cognitive processes undertaken by subjects and potential weaknesses of the methodology when directly applied to the business domain. The second phase of the study develops an object-oriented process modeling formalism based on the results of the first phase.

2. Object-Oriented Process Modeling and Business Engineering

2.1 The Object-Oriented Methodology

The object-oriented methodology abstracts objects and associations between objects to reduce the complexity of a real world system [9][10][11][7][12][13][6]. This provides us with models to help understand, analyze and design a complex system. The object is the basic unit of decomposition in the object-oriented methodology. The object embodies data and procedures encapsulated in itself and the whole system is made up of many such objects that interact to achieve the system's goals. Furthermore the object-oriented methodology provides multiple models that describe the system from various different abstractions. The most commonly provided models of the object-oriented methodology are the static model, the dynamic model and the functional model. These models are represented as diagrams that show views of the system from various perspectives.

The static model represents the static structure of the system. It describes the objects in the system and the relationships between the objects. The static model is represented by class diagrams which describe the structure of the system in terms of classes and associations of classes. The dynamic model describes the aspects of a system concerned with time and the sequencing of operations. It is represented graphically with state transition diagrams (STD) which show the state and event sequences permitted in a system for one class of object, and event trace diagrams (ETD) that show in temporal order the interactions among a set of objects to complete a particular execution of a system. The functional model describes what a system does in terms of transformation of values. It is represented by data flow diagrams (DFD) which show the flow of data values and the transformation of these data values, and input-output diagrams that show the input and output data transformation at a high level.

2.2 The Object-Oriented Process Modeling

Object-oriented process modeling refers to the application of the object-oriented methodology in the analysis and design of business systems. The object-oriented methodology is currently recognized as the promising method for process innovation [14][2][15]. The object-oriented methodology in itself has many advantages for use in business process reengineering. First of all, there is a one-to-one mapping from the problem domain to the solution domain; the business system in the actual world is represented along with the information system in an integral manner [15][14][16]. Likewise, object-oriented design implements the real world into the computer system as 'naturally as possible' [17]. Second, the data and the procedures are grouped into a single construct which promotes simplicity and more efficient communication among the people involved in the project [2]. In addition the object-oriented models easily adapt to change [18], and promotes code reuse [19].

However, compared to the evident advantages and potential of the object-oriented methodology in the software engineering domain, it is not clear that the methodology can also be equally applied to the business domain. For effective application in the business engineering domain a method should support the cognitive tasks of the people analyzing and designing the business system as well as information system. The method should help understand the business processes, identify the core processes, identify critical malfunctions to the core processes and suggest innovative ways to redesign and implement the business system.

2.3 Integration and Divergence of Information

A thorough understanding of the business process is important for business process reengineering. The understanding process involves structuring the meaning of the whole system by integrating information acquired through perception [20]. Especially, in "The Understanding Program" [21], a computer model of the understanding process, human understanding is defined as 'language interpretation' and 'construction'. The language interpretation process consists of extracting important information while reading the description of a system. The construction process involves forming a mental model of the system by integrating the information extracted at the language interpretation phase. However, the interpretation and the construction of the mental model of a business system is difficult due to limits in working memory capacity. The working memory is the information in long term memory that is currently in the temporary memory state of activation [22]. The characteristics of human working memory influences the interpretation and the construction of the mental model of a system. The working memory can only process four to five chunks[1] of information at a specific time. Thus working memory overload occurs when the amount of information a person has to process exceeds the limits of working memory. This leads to a loss of relevant information, which results in failure to construct the mental model of the system [23].

In addition, the method of presenting the information influences the level of understanding. Understanding is easiest when the information is presented in an integrated form that fits the objective of the problem solving strategy, or is in a state of activation in working memory. However, understanding becomes difficult when the information is not activated in working memory and needs to be retrieved from long term memory. Furthermore, understanding is most difficult if the relevant information has not been presented, thus not stored in the long term memory, because the problem solver then has to infer the missing information.

Therefore this study investigates through experiments whether the current object-oriented methodology supports the understanding process for effective business process reengineering. In case the current object-oriented methodology does not support the understanding process, we propose alternative architecture and notations for the methodology to support the understanding process.

3. An Experimental Study of the Effectiveness of Object-Oriented Methodology for BPR

3.1 Subjects

We investigated the effectiveness of the object-oriented methodology for business processes reengineering. Seventeen subjects participated in the study. All subjects who participated in the experiment were junior or senior undergraduate students majoring in business administration at Yonsei University. The subjects were divided into two groups; the ‘model’ group and the ‘control’ group. Nine and eight students were assigned to the ‘model’ group and the ‘control’ group, respectively.

The subjects in the ‘model’ group were students who had taken a semester course in object-oriented analysis and design. These students had no prior experience with real world business engineering projects but were very familiar with the multiple models of the object-oriented methodology and business process reengineering. The subjects in the ‘control’ group were students interested in business process reengineering and management information systems, so they were familiar with business process reengineering concepts. However they had no prior knowledge of the object-oriented methodology

3.2 Experiment Material

The experiment consisted of identifying problem areas to real world business cases. We used the following three criteria for selecting the material. First of all, the business in the case should have severe difficulties with its current processes that only radical innovation and not just simple improvements can rectify the current state. Second, the problem areas in the business should be scattered over multiple perspectives. Thus the multiple models (static, dynamic, and functional models) should have equal importance for analyzing the current state of the business processes. And finally, the case should not be too complex or have a scope too large that the subjects could not possibly solve the problem within the time limits of the experiment. In addition the possibility that the subjects have prior knowledge of the case should be low.

Following the above conditions we selected the ‘Taco Bell’ business case [3][24][1] as the material for our experiment. Taco Bell carried out inefficient work processes in the kitchen of each of their restaurant outlets. This caused severe constraints to their services and profits. Taco Bell reengineered with a system call K-Minus (Kitchenless Restaurant). All the food is cooked outside the restaurant at central commissaries and are just heated at the restaurant when served. This resulted in better service and substantially higher profits. For the purpose of the experiment the ‘Taco Bell’ case was disguised so that the subjects would not recognize it.

With the disguised ‘Taco Bell’ business case we created the material for the experiment. A textual description (Problem Statement) of the case was written for the ‘control’ group. For the ‘model’ group we gave a short introduction (Intr) and multiple diagrams (1 class diagram (CD), 1 event trace diagram (ETD), 5 state transition diagrams (STD), 1 input-output flow diagram (I/O), 3 data flow diagrams (DFD)) of the case modeled following OMT [11]. The problem statement and the models were designed so that they had exactly the same amount of information.

3.3 Experiment Design

At the beginning of the experiment the experimenter explains the objectives of the experiment, the average time of the experiment, and the time limit of ninety

[1] A chunk is a bundle of information that is closely related.

minutes. Afterwards standard warming up procedure for concurrent verbal protocol was undertaken [25][26]. After the warming up session the experimenter provided the subjects with the material. Then the subjects were asked to identify the problems to the business process of the case material.

3.4 Analysis Procedure

For the subjects in the 'model' group the entire experiment process was videotaped and was coded based on concurrent verbal protocols, action protocols and written notes of the subjects [25][26]. The coded protocols were then used to draw the Model Behavior Graphs (MBG). The Model Behavior Graph has a similar structure to the Problem Behavior Graph used by Newell and Simon[27]. It represents the subject's verbal and action protocols while navigating through the multiple models to identify the problem areas as time progresses [28]. The horizontal axis on the Model Behavior Graph represents the time series, while the vertical axis represents the different diagrams provided to the subjects. The actions and verbal utterances were recorded on the Model Behavior Graph according to the time and the diagram the subject observed. We assumed that a subject 'observed' a particular diagram if he paid attention to that diagram for more than one second. Thus even if a subject had a diagram in front of him/her for more than one second, it was not recorded on the Model Behavior Graph if his/her actions or verbal utterances were irrelevant to the diagram in question.

The focus of this study was to investigate how the subject's analyzed the multiple models of the object-oriented methodology. The 'control' group was designed and needed only to compare the results with those of the 'model' group. Therefore for the subjects in the 'control' group we did not use the protocol analysis method because we were not interested in the cognitive processes of the subjects in the 'control' group.

The performance of the two groups was measured by counting the number of subjects that identified the problems that matched the four problems in the initial 'Taco Bell' case. Two different coders coded the answers separately and the results were compared for reliability testing. There were no differences in the results coded by both coders.

3.5 Results

The two groups did not show significant differences in performance. In addition there were two problems no subject could identify. Table 1 shows the number of subjects that identified the correct problems. The problems the subjects had to identify were somewhat different in their inherent nature. Problem #1 and problem #2 were time oriented problems. Problem #1 was that the task of preparing the food material was inefficient and hence took a long time to complete. Problem #2 was that the time to deliver an order was far too long for some menus. Problem #3 was a cost oriented problem. The restaurant outlets were paying a very high rent. This was caused by the large area of the kitchen needed in the outlet that was necessary for preparing the food material and the food ordered. And problem #4 was a quality oriented problem. Each of the outlets were serving the same menu but the taste was not consistent over all the outlets.

The results of Table 1 show that the object-oriented methodology does not support the identification of problems in business processes for business process reengineering. There may be a problem in the process of analyzing the models (i.e. the models may effectively support the analysis of a business process, but the overall understanding of a business process may depend on how a person perceives and utilizes these models).

Table 1 - Performance Differences

	Number of Subjects	Problem #1	Problem #2	Problem #3	Problem #4
Model	9	3	5	0	0
Control	8	4	5	0	0

In order to investigate how the subjects in the 'model' group analyzed the diagrams the subjects were again divided into two groups; a high performance group and a low performance group. The high performance group was assigned the five subjects who identified at least one of the two correct problems (Problem #1 and Problem #2), and the low performance group was assigned the four subjects who had not been able to find any of the correct problems.

3.6 High Performance Group vs. Low Performance Group

In order to compare the differences in cognitive processes between the subjects in the high performance group and the subjects in the low performance group we built model behavior graphs (MBG) for each of the subjects. We will first analyze the model behavior graphs and then examine the differences in the cognitive processes of the subjects in the two performance groups.

3.6.1 Comparison of the Model Behavior Graphs of Two Typical Subjects. Figure 1 and 2 show the model behavior graphs for typical subjects in the two performance groups. The typical subject in the high performance group (Figure 1) examined the introduction (Intr) of the business case from the start of the experiment until 3 minutes 50 seconds after the beginning. Then from 4 minutes 1 second the subject observed the class diagram (CD), a state diagram (STD1), the input-output flow diagram (I/O), then returned to the class diagram (CD) until 9 minutes 10 seconds. Then until 17 minutes 19 seconds the subject examines the event trace diagram (ETD) and the class diagram (CD) alternatively. Then examine a state transition diagram (STD1) until 18 minutes 26 second, when he re-examined the class diagram, then alternatively examined two state transition diagrams (STD1 & STD2) until 20 minutes 47 seconds. Afterwards, he alternatively examine the class diagram (CD), and two state transition diagrams (STD3 & STD4). Then he examines a state transition diagram (STD2) and the input-output flow diagram (I/O). From the input-output flow diagram (I/O), he switches to a state transition diagram (STD5) then looks at a data flow diagram (DFD2) then the class diagram (CD). This sequence took from 26 minutes 32 seconds until 30 minutes 4 seconds. Likewise it took the subject 31 minutes 44 seconds to examine all the diagrams presented.

The typical subject in the low performance group (Figure 2) examined the introduction (Intr) and the class diagram (CD) until 5 minutes 4 seconds. From 5 minutes 6 seconds he examined the event trace diagram (ETD), then refers to the class diagram (CD) for 1 minute 43 seconds. From 9 minutes 17 seconds he alternatively examine two state transition diagrams (STD1 &STD2). From 13 minutes 52 seconds the subject examined two state transition diagram (STD3 & STD2), after which he examined the rest of the diagrams in a sequential manner. It took the subject 17 minutes 26 seconds to examine all the diagrams presented.

We may see from the model behavior graphs that the typical subject in the high performance group actively navigated from and to diagrams in search of relevant information, whereas the typical subject in the low performance group merely examined the diagrams in a sequential manner. Therefore, we compared the cognitive process undertaken by the subjects in the two groups according to the following criteria. First, we compared the number of times the subjects changed the diagram he/she was paying attention to; navigation frequency. Second, we compared the number of diagrams the subject observed before returning to a base diagram; number of intermediate diagrams. And third, we compared the correlation between identifying the correct problem areas and integration of multiple diagrams; integration of multiple diagrams.

3.6.1 Navigation Frequency

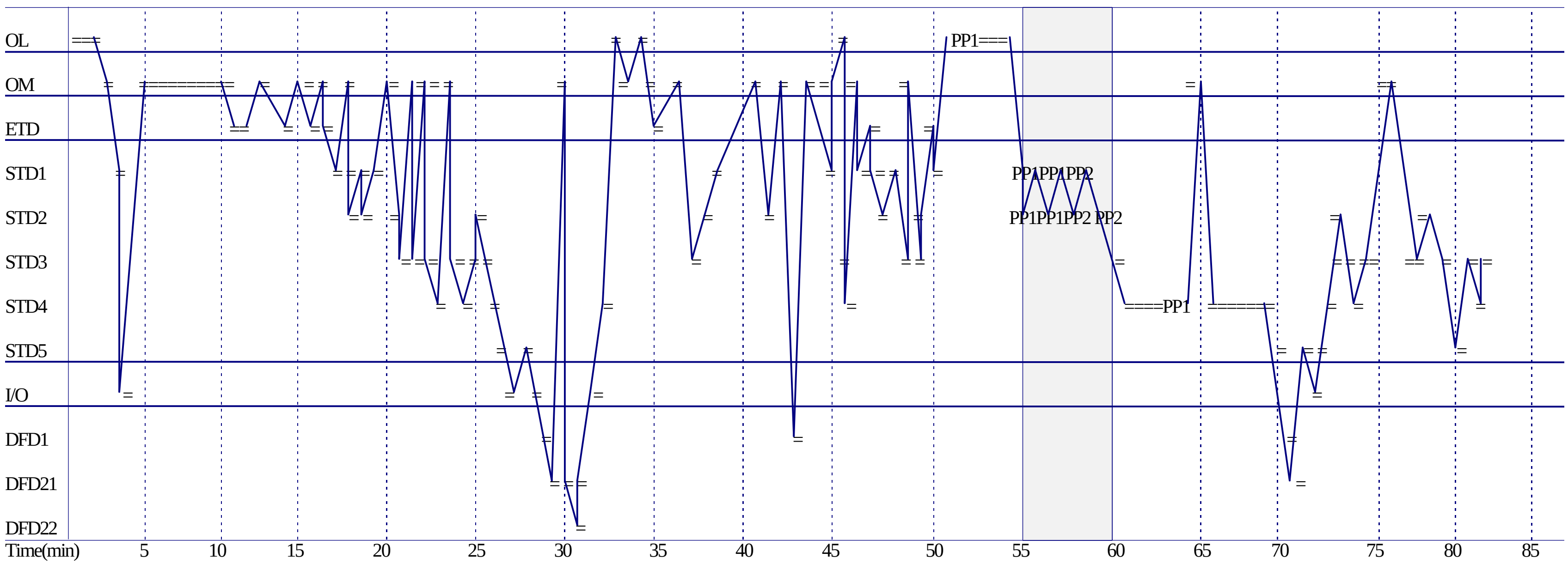
'Navigation Frequency' refers to the number of diagrams a subject observed for more than one second during the experiment. For example if a subject started out with the object model, then changed to a state transition diagram, then another state transition diagram, and after that referred to a data flow diagram, then the navigation frequency for that subject would be 4. Likewise we determined the frequency navigation of all the subjects based on the Model Behavior Graphs. Table 2 represents the navigation frequency of all the subjects in both the high performance group and the low performance group.

Table 2 - Navigation frequency between the high performance group and the low performance group

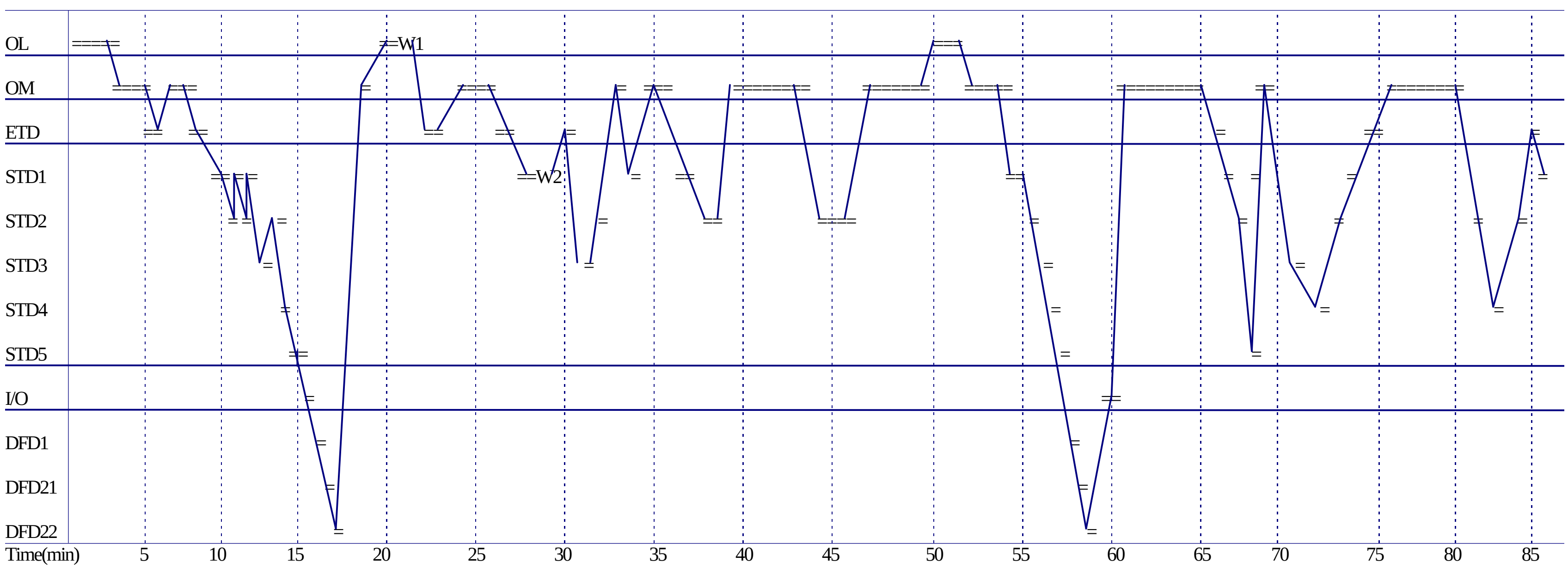
	Subject	Navigation Frequency
High Performance Group	SBJ1	88
	SBJ2	130
	SBJ3	110
	SBJ4	103
	SBJ5	135
	Average	113.2
Low Performance Group	SBJ6	68
	SBJ7	87
	SBJ8	105
	SBJ9	65
	Average	82
t - test	df = 7	t(7)=2.52

According to Table 2 all the subjects in the high performance group except SBJ1 scored a navigation frequency of over one hundred times. Furthermore all the subjects in the low performance group except SBJ8 scored a navigation frequency of less than one hundred times. Therefore we can conclude that the subjects in the high performance group changed diagrams many more time than those in the low performance group ($t(7) = 2.52$, $p < 0.05$).

3.6.2 The Number of Intermediate Diagrams. The number of intermediate diagrams refers to the number of diagrams a subject paid attention to before returning to a base model in order to integrate the information dispersed over multiple diagrams into a single. The criteria for comparing the number of intermediate diagrams was determined according to limits in human working memory capacity. Table 3 and Figure 33 shows the frequency of navigation until returning to a base diagram with the number of intermediate diagrams



<Figure 1> Model Behavior Graph of Typical Subject in the High Performance Group



<Figure 2> Model Behavior Graph of Typical Subject in the Low Performance Group

equals to 1 through 10.

Table 3 - Number of Intermediate Diagrams Between High Performance Group and Low Performance Group

Number of intermediate diagrams	1	2	3	4	5	6	7	8	9	10
High Performance Group	171	33	23	4	20	23	13	18	9	20
Low Performance Group	68	16	11	19	22	11	11	8	8	5

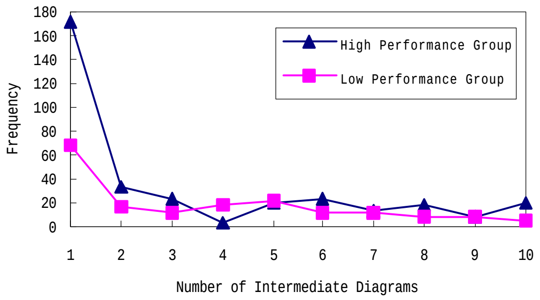


Figure 3 - Navigation Frequency by Number of Intermediate Diagrams

We can see in Table 3 and Figure3 that when the number of intermediate diagrams is less than 3 (i.e. the total number of diagrams observed is equal to 4) the subjects in the high performance group scored a higher frequency of navigation than the subjects in the low performance group. Particularly, when the number of intermediate diagrams is equal to 1, the results show that the subjects in the high performance group significantly outscored the subjects in the low performance group ($t(7) = 2.68$, $p < 0.05$). When the number of intermediate diagrams exceeds 4 the navigation frequency for the high performance group decreased considerably to 4 while the subjects in the low performance group sustained a high frequency of 19.

When the number of intermediate diagram exceeds 4, the total number of diagrams referred to becomes 5, the amount of information may surpass the limit of working memory capacity. Therefore we can assume that when the number of intermediate diagrams exceeds 4, information from the diagrams referred to do not undergo successful cognitive integration. We can see in Table 3 and Figure 3 that the navigation frequency for both performance groups show somewhat irregular trends when the number of diagrams was over 4, which implies that the subjects may have been just browsing through a large number of diagrams. Furthermore the navigation frequencies for the low performance group are quite stable across all numbers of intermediate diagrams over 2. These frequencies showed no fluctuation and no particular trends.

Hence, in summarizing evidence relating to the navigation frequency, we see that the subjects in the high performance group do not merely navigate over many diagrams, but take on a cognitive strategy that integrates information from multiple sources where the information do not overload the limit of working memory capacity. The subjects in the high performance group started at a base diagram and sought information from a number of other diagrams and came back to the base diagram in order to integrate the information sought. All this was done while keeping the number of intermediate diagrams below the limit of working memory capacity.

3.6.3 Correlation Between Integration of Multiple Diagrams and Performance. In the previous sections we found that cognitive integration was successful when the number of diagrams referred to before returning to a base diagram was between one and three. In this section we related the cognitive integration with the actual performance. For this we investigated if the subjects went through cognitive integration of the diagrams while identifying the problems to the given case material. Table 4 shows whether cognitive integration of multiple diagrams took place while identifying the problems to the case material. The O before the arrow (->) means that cognitive integration took place, and the O after the arrow means the problem was identified at the point in time when the cognitive integration was terminated. The results are not applicable for problem #1 of SBJ1 and SBJ5 because these subjects did not correctly identify the problem.

Table 4 - Cognitive Integration for Problem Identification

Group	Subject	Problem Identification with Cognitive Integration	
		Problem #1	Problem #2
High Performance Group	SBJ1	N/A	O -> O
	SBJ2	O -> O	O -> O
	SBJ3	O -> O	O -> O
	SBJ4	N/A	O -> O
	SBJ5	O -> O	X -> O

We can see in Table 4 that all the problems identified (except for problem #2 of SBJ5[2]) followed cognitive

[2] Even though problem identification for problem #2 of SBJ5 did not promptly follow cognitive integration, SBJ5 showed integration of the information relevant to problem #2 during the experimental period, so we may assume that prior cognitive integration helped the subject to understand the problem domain and had a positive impact on problem

integration. These results imply that cognitive integration is necessary for understanding the business system and identifying the problems from this understanding.

To summarize the results of the experiment, we found that in order to gain comprehensive understanding of a business system the subjects went through a process of cognitive integration where he/she navigated frequently from and to diagrams in search of information relevant to a particular business process. However, the subject kept the number of referring diagrams under the limit of working memory capacity in order for working memory overload not to occur. In addition the subjects had difficulty forming a mental model of the business process which he/she needed in order to assess whether the process under question had problems or were inefficient. Moreover, there were two problems in the business case (problem #3 and problem #4) no subject could identify.

4. An Object-Oriented Process Modeling Method for Business Engineering

The results of the experimental study discussed previously suggest directions for developing an object-oriented process modeling method to support business process reengineering. The methodology needs to support the understanding process by providing information integrated from a process perspective rather than from the static, dynamic and functional perspectives which do not properly fit the objective of the problem solving strategy. Thus, the methodology should provide a formalism for providing views of the business processes and also a mechanism for capturing the richness of the business domain such as cost and quality issues that are crucial for the understanding and assessment of the goodness of a business process.

The formalism we are proposing is based on the findings of the experiments and on the notation of the 'Unified Modeling Language[3]' [29], which we believed would have a great influence on the object-oriented community. The 'Unified Modeling Language' is a method for specifying, visualizing, and documenting the artifacts of an object-oriented system under development.

The results of the experiment propose that, for application in the business domain, diagrams should provide explicit views of business processes. Moreover these views should exhibit a variety of properties of the business process that are crucial to the understanding of the processes and also to asses whether a business process under consideration embodies critical problems. We will first discuss the development of the process views and then the multi-dimensional view.

4.1 The Process View

The 'Unified Modeling Language' provides various diagrams that were not presented to the subjects at the time of the experiment. These diagrams are the use case diagram, the message trace diagram and the object message diagram. The use case diagrams describes the behavior of a set of objects. It shows how actors external to the system interact with the system and use the system. The use case diagram provides an outside view of the system. The message trace diagram shows the interaction among a set of objects in temporal order, which is good for understanding time issues. The object message diagram shows the interaction among a set of objects as nodes in a graph, which is good for understanding the system structure.

In the business domain use cases can be regarded as business processes; the processes that serve the actors external to the business system by providing them with some value. The use case diagram describes the behavior of a set of objects. A use case is a generic description of an entire transaction involving several objects. The use case may have instances which are called scenarios. Scenarios illustrate interactions among objects to achieve the behavior of the use cases. These scenarios are depicted in detail in the form of message trace diagrams or object message diagrams, where the message trace diagram emphasizes the temporal order of interactions and the object message diagram emphasizes the structural relationships among the objects. Thus the use case diagram can represent the business processes in an aggregate form, and the message trace diagrams and the object message diagrams can show how these business processes are processed in terms of interaction among objects. Therefore the use case diagram, message trace diagram and the object message diagrams are very important for understanding the system, because they are based on scenarios which are the instances of use cases. Hence these diagrams have the potential of providing the process views that was implicated by the results of the experiment.

The use case diagrams, message trace diagrams and the object message diagrams seem to provide the process view for the business system, but have not been directly applied to business engineering. The use case diagram in the unified modeling language only show the business processes at a very high level. These are useful for identifying business processes but are insufficient at

identification.

[3] Major methodologists using the object-oriented approach have united to develop the 'Unified Modeling Language' which strive to provide a standard methodology for the development of object-oriented information systems.

providing the details for minute understanding of the business processes. Use case diagrams should be specified at a lower level to provide more detailed understanding of the specifics of the business processes. In the object-oriented process modeling method we are proposing, we have added a diagram called the 'Low-level Use Case Diagram'. The low-level use case diagram uses the same notation as the use case diagram and shows from an inside perspective the structure and the dynamics of the use cases and also the dependencies among use cases. The low-level use case diagram incorporates the 'uses' and 'extends' relationships among use cases as in the unified modeling language [15]. The 'uses' and 'extends' relationships show the structure of the use cases. We have also added other relationships to show the dynamics and dependencies of and among the use cases. The 'flow' relationship among the use cases indicate that one use case follows another. It is represented by an arrow to show the sequence of use cases. The 'depends on' relationship among use cases indicate that a use case is a precondition to the execution of another use case. It is represented by a dashed arrow from the 'depender' use case to the 'dependee' use case with the keyword 'depends on' an or adjacent to the dashed arrow.

The additional constructs we have included are mapped from the object message diagrams to the low-level use case diagram. The object message diagram often manifests very complex behavior; the scenario may be composed of a number of sequential subroutines, and may also include concurrency of interactions and preconditions to operations. The sequence of operations of the object message diagram are mapped onto the low-level use case diagram as a 'flow' of use cases. The concurrent processes are mapped as multiple use cases that 'flow' from a particular use case. The preconditions to operations are mapped as dependency relationships.

4.2 The Multi-Dimensional View

In the previous section we have proposed a way to model the business process from a 'process perspective'. The low-level use case diagram we have developed provides a process view of a business system and offers constructs that allow to model the complexity inherent in the business process. The low-level use case diagram shows the structural, procedural and dependency relationship among use cases. These aspects allows the understanding at a meso level; one can see what activities are performed, in what order they are processed, and that some activities depend on other activities in order to proceed.

However such a 'workflow' model only provides a superficial understanding of the business process; it only offers a model that describes what the business does and how they do it in terms of activity flow and dependency. Now that we have an idea about what is being done, we need to find out what's wrong with what our business is doing. In other words the use case diagrams help to identify the business processes, and the low-level use case diagrams help to gain an understanding of how the business processes are constituted. The next steps are identifying the core business processes that will be considered for innovation, and also the problems in these core processes that need to be innovated. The process modeling method should support these tasks.

In this study, we are proposing to address these issues with the message trace diagrams. As mentioned above, the message trace diagram describes a scenario in terms of a sequence of messages or interactions. Messages that take time can be drawn as arrows slanted downwards and timing marks can explicitly be expressed on the diagrams to provide more detail. However the message trace diagram only addresses temporal aspects of a scenario; how long an activity takes, and how long an object processes maintains a state, etc. The business domain is consists of more than just time driven issues. There are other factor in the domain that are important and need to be considered. Cost and quality are such factors. We need to identify and question the activities that entail high cost, or activities that have a negative impact on the overall quality of the products and services of the business.

We propose to model these factors on the message traces diagrams just as time issues were expressed. The message trace diagram thus expresses a scenario, with time, cost and quality marks explicit on the interactions or on states. These multiple factors can be expressed all at once on the diagram or once per diagram. Tools should be able to provide such functionality with filtering functions to let the analyst choose the amount of information he/she desires.

5. Conclusion & Discussions

This experimental study examined the effectiveness of the object-oriented methodology in understanding the current business system and its processes for business engineering. The results of the experiment propose guidelines for the development of process modeling methods. First, the results of the subjects' cognitive processes show that a vast amount of information needs to be integrated in order to gain a comprehensive understanding of business processes. The current object-oriented methodologies abstract systems from multiple perspectives which is necessary for the design and implementation of information systems, yet the methodologies are not appropriate when we need to identify and gain profound understanding of the core processes of the business system. Diagrams explicitly providing a process view would spare the cognitive burden of having to construct a mental model of the business system in order to integrate the vast amount of information into such a process view.

Furthermore, the results of the experiment tell us that even the subjects that successfully constructed the mental models could only identify problems that were time related. This is due to the fact that current object-oriented methodologies only describe the dynamics of a system in terms of time. Thus modeling and understanding business processes requires diagrams that encourage the thinking of the business process from multiple dimensions in addition to the temporal dimension. Consequently, object-oriented process modeling methods need to support to recognition of the core processes of a business system from multiple dimensions such as time, cost and quality, and diagrams that represent these core processes need to be developed. This study proposes such a process modeling method. The method uses the notation from standard object-oriented methodologies and provides additional constructs to describe a business process with information from multiple dimensions such as time, cost and quality. The method also supports the process of identifying the core processes.

In sum, This study proposes guidelines for the development of an object-oriented process modeling formalism that supports the cognitive task of the business engineer when analyzing and understanding business systems . However, our works have several limitations. First, the number of subjects that participated in the experiment was not large enough to firmly advocate the results of this study. However, this study applied a process tracing method of analysis and focused on the qualitative aspects of the subjects cognitive processes instead of on the quantitative aspects of the final performance results. Second, we used 'OMT' [11] that is very much focused on the design and development of information systems. Other object-oriented methodologies such as 'Objectory' [15] and 'OOAD' [7] provide constructs that may be more suitable for the business domain. At the time of the experiment the three methodologies (OMT, Objectory, and OOAD) coexisted, and no one methodology was proven more effective for business process reengineering. We judged that the notation of OMT was sound enough for the purpose of the experiment. Third, the effectiveness of the object-oriented methodology was investigated through laboratory experiments with a single business case. Therefore, the results may not be generalized to the actual business engineering practice.

The above limitations suggest areas for future work. The effectiveness of the proposed method needs to be tested with a larger variety of business cases both empirically through laboratory experiments and field tests. Furthermore, the method should be expanded to support the innovation process once the business processes are thoroughly understood.

References

[1] Hammer, M., & S. A. Stanton, *The Reengineering Revolution*, Harper Business, New York, 1995

[2] Davenport, T. H., Process Innovation: Reengineering Work Through Information Technology, Harvard Business School Press, Boston, 1993.

[3] Hammer, M. & J. Champy, Reengineering the Corporation, Harper Business, New York, 1993.

[4] Davenport, T. H. & J. E. Short, The New Industrial Engineering: Information Technology and Business Process Redesign. *Sloan Management Review*, Summer 1990, pp.11-27.

[5] Van Meel, J. W., The Dynamics of Business Engineering: Reflections on Two Case Studies Within the Amsterdam Municipal Police Force, Doctoral Dissertation University of Technology Delft., 1993.

[6] Jacobson, I., Object-Oriented Software Engineering: A Use Case Driven Approach, Addison Wesly, Reading, Massachusetts, 1992.

[7] Booch, G., Object-Oriented Analysis and Design with Applications, 2nd Ed., Benjamin/Cummings Publishing Company, California, 1994.

[8] Winograd, T., From Programming Environments to Environments for Designing, Communication of ACM, 1995, 57-66.

[9] Cox, B. J., Object-Oriented Programming: An Evolutionary Approach, Addison Wesly, Reading, Massachusetts, 1987.

[10] Meyer, B., Object-Oriented Software Construction, Prentice Hall, New Jersey, 1988.

[11] Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy & W. Lorensen, Object-Oriented Modeling and Design, Prentice Hall, New Jersey, 1991.

[12] Yourdon, E., Object-Oriented Systems Design: An Integrated Approach, Yourdon Press, New Jersey, 1994.

[13] Martin, J. & J. J. O'dell, Object-Oriented Methods: A Foundation, Prentice Hall, New Jersey, 1995.

[14] Taylor, D. A., Business Engineering with Object Technology, John Wiley & Sons. Inc., New York, 1995.

[15] Jacobson, I., The Object Advantage: Business Process Reengineering with Object Technology, Addison Wesly,

Reading, Massachusetts, 1995.

[16] Kim, J. & F. J. Lerch, "Toward a Model of Cognitive Process in Logical Design: Comparing Object-Oriented and Traditional Functional Decomposition Software Methodologies", *Human Factors in Computing Systems : CHI Proceedings*, May 1992, pp.489-498.

[17] Kim, J., F. J. Lerch & H. A. Simon, "Internal Representation and Rule Development in Object-Oriented Design", *ACM Transactions of Computer Human Interaction (TOCHI)*, 1995.

[18] Wang, S., "OO Modeling of Business Processes: Object-Oriented Systems Analysis", *Information Systems Management*, Spring 1994, pp.36-43.

[19] Beedle, M. A., "Object-Based Reengineering", *Object Magazine*, March-April, 1995, pp.53-58.

[20] Carpenter, P. A. & M. A. Just, "Cognitive Processes in Reading". In J. Orasanu (Ed.), *Reading Comprehension: From Research to Practice*, Lawrence Erlbaum, Hillsdale, New Jersey, 1986.

[21] Simon, H. A. & J. R. Hayes, "Understanding Written Problem Instructions", Models of Thought, Vol. 1, 1979, pp.451-476.

[22] Barsalou, L., *Cognitive Psychology: An Overview for Cognitive Scientists*, Hillsdale, New Jersey: LEA Publisher, 1992

[23] Anderson, J. R. & R. Jeffries, "Novice LISP Errors: Undetected Loss of Information from Working Memory", *Human Computer Interactions*, Vol.1, 1985, pp.107-131

[24] Champy, J., Reengineering Management: The Mandate for New Leadership, Harper Business, New York, 1995.

[25] Ericsson, K. A. & H. A. Simon, Protocol Analysis: Verbal Reports as Data, MIT Press, Cambridge, Massachusetts, 1993.

[26] Van Someren, M. W., Y. F. Barnard, & J. A. C. Sandberg, The Think Aloud Method: A Practical Guide to Modeling Cognitive Processes, Academic Press, San Diego, 1994.

[27] Newell, A. & H. A. Simon, Human Problem Solving, Prentice Hall, New Jersey, 1972.

[28] Kim, J., K. Yi & J. Hahn, "Put Yourself in Somebody Else's Shoes: Representation Change in Primal-Dual Linear Programming", *Asian Pacific Decision Science Institute, accepted*, 1996.

[29] Booch, G. & J. Rumbaugh, "Unified Method for Object-Oriented Development", Rational Corporation, 1995.