Experimenting with Real-Time Specification Methods: The Model Multiplicity Problem

Mor Peleg and Dov Dori

Area of Information systems Engineering
Faculty of Industrial Engineering and Management
Technion, Israel Institute of Technology, Haifa 32000, Israel
{mor, dori}@ie.technion.ac.il; Fax 972-4-8235194 , Tel: 972-4-8294409 / 2921

The Object-Process Methodology (OPM) specifies both graphically and textually the system's static-structural and behavioral-procedural aspects through a single unifying model. This model singularity is contrasted with the multi-model approach taken by existing object-oriented system analysis methods. These methods employ several models for specifying various system aspects – mainly structure, function, and behavior. Object Modeling Technique (OMT), the main ancestor of Unified Modeling Language (UML), extended with Timed Statecharts, represents a family of such multi-model object-oriented methods. Two major open questions have been whether or not a single model, rather than a combination of several models, enables better system specification, and which of the two alternative approaches yields a specification that is easier to comprehend. In this study, we address these questions through a controlled experiment, and establish empirically that a single model methodology – OPM – is more effective than a multi-model one (OMT) in generating better system specification. The paper also discusses the significant differences between OPM and OMT that were found in specification comprehension.

1. Introduction

Object-Oriented methods are being widely used for system analysis and specification. The majority of these methods advocate the use of a combination of several models – usually a static-structural, or object model, a functional model, a behavioral model, and a host of additional optional models – to describe the various aspects of the system under consideration. The use of multiple models to describe a system from various aspects is the cause for the *model multiplicity problem.* The problem is that, to get a real, deep comprehension of the system being studied and the way it operates and changes over time, various models have to be referred to concurrently. Real-time requirements exacerbate this problem even further, as either yet another model needs to be introduced to reflect the additional temporal demands, or they are added into one of the existing models (usually the dynamic/behavioral model).

Due to the fact that multiple models are employed in order to specify a system, model incompatibility problems at both the syntactic and the semantic levels are likely to show up. Furthermore, the integration of the different models that describe different aspects of the system into a coherent unity is seldom explicit. Hence, the burden of mental integration is put on the shoulders of the analyst and his/her target audience. Technical solutions that involve sophisticated CASE tools, advanced as they may be, to impose consistency, may alleviate manual consistency maintenance, but they do not address the core problems of information overwhelming and mental integration with which humans are faced. Since the required fusion of the various models within one's mind is extremely difficult, our conjecture has been that it may adversely affect the two transformations involved in the development and use of information systems [1]: the representation of the real-world system as a system specification and the comprehension (interpretation) of the system specification. Indeed, our empirical study, described in this research, supports this claim.

We compared OPM/T [2] (OPM [3] with temporal extension) – a graphic object-oriented system specification method, which is unique in its ability to specify the structural, functional and dynamic aspects of systems in a single model, with T/OMT (Timed OMT) - a temporal extension of OMT [4-6], which was selected as a representative of multiple-model methods. The core of this research is a controlled experiment, aimed at demonstrating the model multiplicity problem through a comparison between the two methods. For the sake of brevity we refer to the two methods compared as OPM and OMT rather than OPM/T and T/OMT. The rest of this paper is structured as follows. Section 2 describes the research design and hypotheses. The results of the experiment are presented in Section 3 and discussed in Section 4. A summary concludes the paper. Appendix A provides the formal problem statements of the case studies used in the experiments.

2. Research Design and Hypotheses

As noted, we compared OPM and OMT in terms of two crucial issues: (1) specification comprehension and (2) specification generation quality. We carried out the experiment within the final examination of a System Analysis and Specification course, described below. In this section we present the research population and the two experiment parts, which we call Specification Comprehension and Specification Quality.

2.1. Population Background and Training

The research population consisted of 88 third year Information Systems Engineering undergraduate students in the Technion, Israel Institute of Technology. The experiment took place during the academic year 1997, at the end of the Systems Analysis and Specification course, which follows a series of mandatory computer- and information-related courses, including Introduction to Computers and Design and Implementation of Information Systems. The course covered, among other topics, system development life cycle and the two system analysis and specification methods that we compared in this study – OMT and OPM, including their temporal extensions. The same instructor (this paper's second author) taught the whole class both OMT and OPM, while two Teaching Assistants (one of whom is this paper's first author) conducted the class recitations in four separate groups. Each Teaching Assistant taught the members of his/her group both methods. OMT was taught first, for four weeks, then OPM was taught for three weeks. Next, OPM/T was taught for two weeks, then T/OMT was taught for two more weeks. The students were required to submit solutions to four exercises, one on each of the methods — OMT, T/OMT, OPM and OPM/T. They took a midterm examination that covered only OMT, a fact that might have had an effect of better performance with OMT relative to OPM due to better preparation.

2.2. Experimental Setting

The independent variable in the experiment was the analysis and specification method used (OPM or OMT), while the two controlled variables were participants and tasks. Our null hypotheses (H_0) were that a difference between the two modeling methods exists neither in Specification Comprehension nor in Specification Quality. For each of the above null hypotheses, there is an opposite (H_1) two-sided hypothesis.

The experimental procedure, presented by the Object-Process Diagram (OPD) of Figure 1, was as follows. For the test, the Class was randomly divided into two Groups consisting of 47 and 41 students each. In the Specification Quality part of the experiment, the participants of both groups received the same problem statement. The participants of the first group were asked to specify the system described textually in the problem statement using OMT, while the participants in the second group were asked to specify the same system using OPM. The use of scaling (zooming) mechanisms of OPM or DFD (OMT's Functional Model) was required in the solutions. The quality of the resulting specifications was thoroughly analyzed, as explained in subsection 3.2.

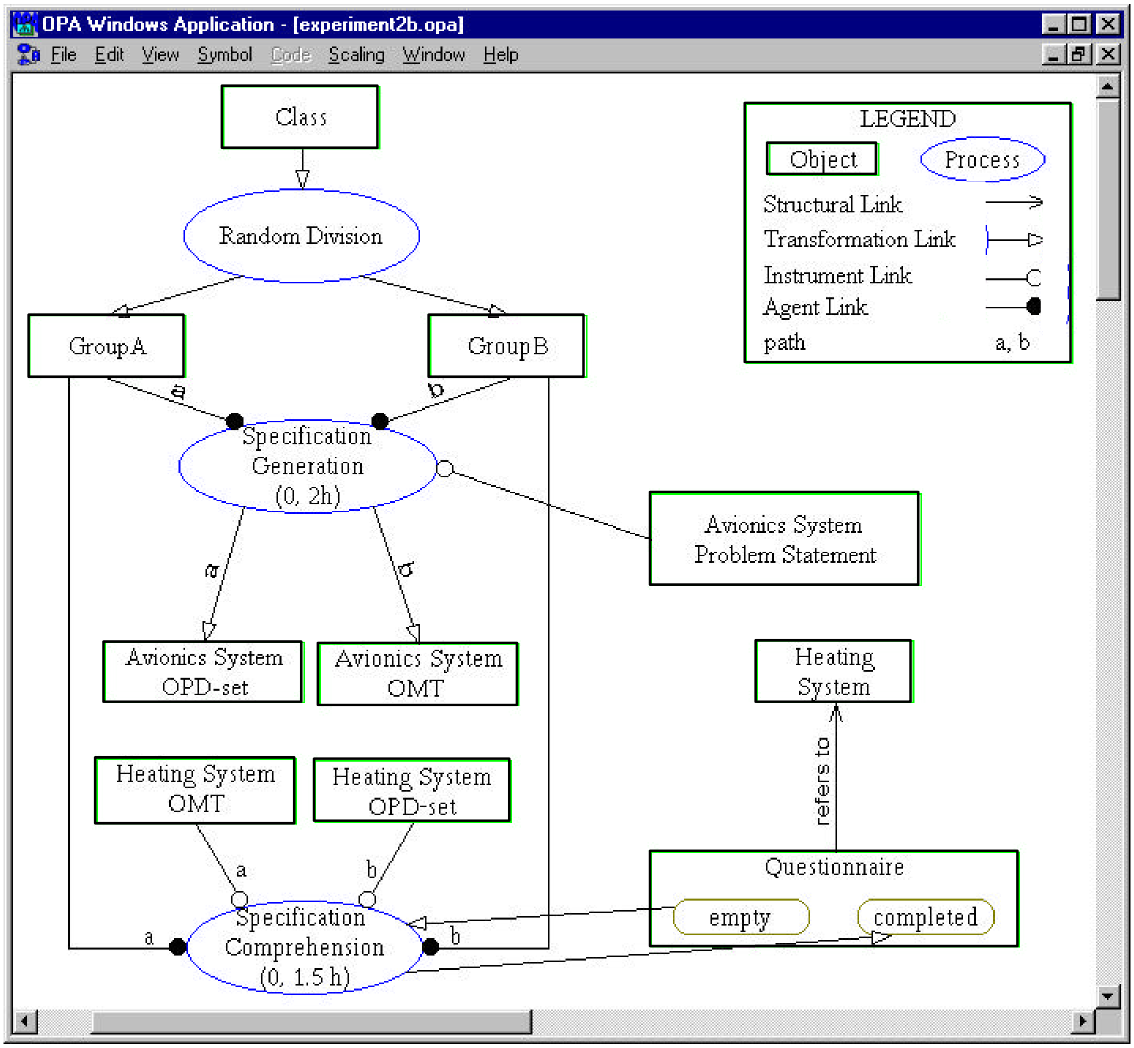

Figure 1. An Object-Process Diagram (OPD) describing the experimental procedure. The OPD shows three processes: Random Division, Specification Generation, and Specification Comprehension. Two different kinds of links are used to connect objects to processes, depending on the roles that these objects play in the process to which they are linked. Objects may serve as enablers – instruments or intelligent agents, which are involved in the process without changing their state. An enabling link is denoted either by ——○ , in the case of an instrument, such as the Avionics System Problem Statement or by ——● , in the case of an intelligent agent, such as the groups of students: Group A and Group B. Objects may also be transformed as a result of a process acting on them. In this case, the transformation link, denoted by an arrow is used. The direction of the arrow specifies whether the transformation is a state change of an object, a generation of an object, or a consumption of an object. The letters 'a' and 'b', depicted above some of the links, mark paths. For example, paths 'a' and 'b' of the process Specification Generation specifies that Group A students produced an Avionics Systems OPD-set, while Group B students produced an Avionics System OMT Diagram-set.

For the Specification Comprehension part, the specification methods were swapped: The participants in the first group, who were requested to specify the textual problem statement using OMT, were now asked to demonstrate comprehension of a system specification provided in OPM. Likewise, the participants in the second group, who were requested to specify the textual problem statement using OPM, were now asked to demonstrate comprehension of a specification provided in OMT. The OMT and OPM specifications used in the Specification Comprehension experiment were equivalent. This paper's first author prepared the two system specifications. An attempt was made to obtain the best solution for each specification method. Two experts verified the solutions; one is this paper's second author, and the other – a teacher assistant in the Systems Analysis and Specification course.

The comprehension of both specification types (OPM and OMT) was determined by the students' response to a questionnaire. Students of both groups received the same questionnaire. The method swapping between the two parts of the experiment canceled out any potential bias in student competence, although the division into the groups was made randomly in the first place.

The students were allowed to use any written material, including class notes. We allocated an hour and a half for the Specification Comprehension part of the experiment, and two hours to complete the Specification Quality part. This time allotment proved to be sufficient for all the students.

At the end of the experiment, we passed a post-test questionnaire in which we asked the students to state their preference between the two methods, according to a 7-point scale, ranging from absolute preference for OPM to absolute preference for OMT. The students were told that they must answer the questionnaire, but that their answers will not be considered as part of their grade in the course.

2.3. Specification Comprehension

For the Specification Comprehension part of the experiment, we used two equivalent, alternative pre-prepared diagrammatic system specifications of the same real-time Home Heating case study, based on Toetenel and Katwijik [7], one in OPM and the other in OMT. The problem statement of the Heating System case study is given in Figure A-1 of Appendix A. Each participant was asked to respond to a questionnaire consisting of 33 closed questions concerning the specifications. We classified the questions according to different specification issues. These issues, along with exemplary questions, are listed in the first two columns of Table 1. The purpose of this classification was to pinpoint specific issues (if any) in which a significant difference between the methods exists. Our set of null hypotheses (H_0) were that there is no difference between OMT and OPM in any one of the comprehension issues and that there is no overall difference between OMT and OPM.

2.4. Specification Quality

In the Specification Quality part of the experiment, both groups were given the same textual problem statement. Using the problem statement, the task of each student was to specify a real-time system in one of the methods he/she was told to use. The problem statement, shown in Figure A-2 of Appendix A, was adapted from Laplante [8]. Following the experimental design of Shoval and Frumermann [9], we classified the different error types into different categories. Checking the specifications the students came up with, we classified the errors we found into 38 error types. The first column of Table 3 lists the 38 error types.

Due to differences in facilities the methods provide, some of the error types could only occur in one of the methods and not in the other. Thus, confusion between an external entity and a datastore and confusion between an activity and an action could only occur in OMT and not in OPM, while confusion between an effect link and an instrument link could only occur in OPM and not in OMT. For these error types we hypothesized that there would be a difference in the number of participant errors between the two methods. For the rest of the error types, we hypothesized that there would be no difference in specification quality between the methods.

Table 1. Exemplary questions and results of the Specification Comprehension part of the experiment for multiple-question issues

Issue	Exemplary question	i = # of Questions Answered Correctly	# of Students who Gave i Correct Answers in OMT	# of Students who Gave i Correct Answers in OPM	Test Statistic	p-Value (two-tailed)
1.Identifying structural relationships	The combustion sensor senses combustion of what object?	0	0	0	-0.60	0.55
		1	0	0		
		2	4	2		
		3	14	16		
		4	23	29		
2.Identifying object states	What are the states of the Fuel Level Indicator?	0	1	0	-0.55	0.58
		1	0	0		
		2	6	5		
		3	34	42		
3.Identifying triggering events of a process	What is (are) the event(s) that trigger the Circulation process? Specify the event(s) and their associated reaction time constraints.	0	2	0	-5.13	<0.001 OPM
		1	11	0		
		2	13	8		
		3	8	6		
		4	7	33		
4.Identifying processes triggered by an event	What process is triggered by the event of Monitored Room Temperature entering a state of >H_d+2?	0	0	0	-2.02	0.04 OPM
		1	3	0		
		2	1	0		
		3	10	5		
		4	27	42		
5.Identifying events that affect objects by changing their states	What events cause Oil Valve to close, and with what timing constraints?	0	0	5	4.54	<0.001 OMT
		1	3	11		
		2	7	20		
		3	16	6		
		4	15	5		
6.Identifying objects that are affected by an event	When the fuel level indicator enters the empty state, an event is registered. What objects are affected by that event, and after how much time from the event occurrence?	0	0	0	0.39	0.70
		1	0	0		
		2	1	4		
		3	8	8		
		4	32	35		
7.Identifying objects that participate in a process	What object(s) participate in the room temperature monitoring process but are not outputs of that process? List objects that serve as data sources, create events that invoke Room Temperature Monitoring process, or participate in this process without contribution of data.	0	3	1	-5.36	<0.001 OPM
		1	18	2		
		2	12	9		
		3	5	12		
		4	3	23		
8. determining relativity of processes execution order	Can Motor Monitoring process start before the Room Temperature Monitoring process starts?	0	0	0	2.08	0.04 OMT
		1	1	0		
		2	0	4		
		3	7	17		
		4	33	26		

3. Results

In this section we present the statistical tests that were deployed in our research and their results.

3.1. Specification Comprehension Results

The level of comprehension was measured for each issue separately by counting for each student the number of (fully) correct answers. For the multiple-question issues we applied the Wilcoxon-Mann-Whitney test [10] — a non-parametric test, used to test whether two independent groups have been drawn from the same population, which is not necessarily distributed normally.

Table 1 lists exemplary questions and results of the Specification Comprehension test for multiple-question issues. The issue names and exemplary questions are given in the first two columns. The third column shows the possible number i (i = 0..number of questions in issue) of correct answers that can be given by a student for each issue. The fourth and fifth columns give the students who gave i correct answers in an issue, using OMT and OPM, respectively. The column before last gives the value of the test statistic. Negative values of the test statistic reflect an advantage of OPM, while a positive value reflects an advantage of OMT. The right-most column shows the p-Values, and in cases of significant differences, the name of the favored method. We considered results with p-Values ≤ 0.05 to be significant.

For the single-question issues—timing exceptions and cyclic processes—we used the test for equality of proportion to determine whether there is a significant difference in the performance on these questions between the two methods. No significant difference between the two methods was found for either one of these two issues. The results are given in Table 2.

Table 2: Summary of results of the specification comprehension experiment for single-question issues

Issue	Proportion of Students that answered correctly in OMT	Proportion of Students that answered correctly in OPM	Test Statistic[1]	p-Value (two-tailed)
Identifying timing exceptions	36/41	44/47	-0.95	0.34
Identifying cyclic processes	40/41	46/47	-0.01	0.99

3.2. Specification Generation Quality Results

We used the test for equality of proportion [11] to determine whether there is a significant difference between the two specification methods in the number of students who made specification errors of various types. The results are summarized in Table 3, where we show the proportion of errors for each error type in each model, along with the value of the test statistic and the p-Value. Results with p-Values > 0.09 are assumed to be non-significant. As Table 3 shows, in 10 out of 38 error types, OPM produced significantly less modeling errors, while OMT produced significantly less modeling errors in only 2 out of these 38 error types.

We used the T-test to measure the difference of the mean number of error types per student for each between the two method types, OMT and OPM. Error types that could only occur in one of the methods were left out of the overall calculations. The results are shown in Table 4. In the first row, we have assumed that all error types have the same default weight. The results show an overall advantage of OPM over OMT. Changing the weights of different error types affects the mean scores of the two methods. Therefore, under certain choices of weights per error types, OPM may not be superior to OMT in the overall specification quality, as measured by the mean number of error types per student. The second row of Table 4 shows that even if we increase by a factor of 4 the weights of all the error types for which OMT had a better score than OPM (even if not a significant advantage), and leave the weights of the other error types unchanged, then OPM still has a better overall score than OMT.

Table 3. Summary of the significant results of the Specification Quality Part of the Experiment

Error Type	Proportion of errors in		test statsitic	Significant difference in favor of:	p-Value (two-tailed)
	OMT	OPM			
1.1 missing link	22/47	22/41	0.64		0.52
1.2 incorrect link	5/47	5/41	0.23		0.82
2.1 wrong multiplicity constraint	1/47	0/41	-0.94		0.35
3.1 using specialization instead of aggregation	0/47	1/41	1.08		0.28
4.1 irrelevant objects	1/47	0/41	-0.94		0.35
4.2 missing objects	16/47	10/41	-0.99		0.32
4.3 overlapping objects/ common methods	2/47	0/41	-1.34		0.18
4.4 an attribute defined as a top-level object	1/47	0/41	-0.94		0.35
4.5 confusing a datastore with an external entity	6/47	0/41	-2.37	OPM	0.03
5.1 missing feature	36/47	16/41	-3.58	OPM	<0.001
5.2 attributing features to irrelevant objects	3/47	0/41	-1.65		0.1
6.1 cyclic processes not modeled as cyclic	9/47	4/41	-1.24		0.22
6.2 mutually exclusive processes modeled as parallel processes	9/47	0/41	-2.96	OPM	0.003
6.3 irrelevant input/ output	11/47	6/41	-1.04		0.30
6.4 missing input/output	24/47	16/41	-1.13		0.26
6.5 overlapping processes	1/47	0/41	-0.94		0.35
6.6 missing processes	6/47	2/41	-1.28		0.20
6.7 confusion between an effect link and an instrument link.	0/47	17/41	4.92	OMT	<0.001
6.8 duration constrained process not modeled as such	5/47	1/41	-1.52		0.13
6.9 wrong aggregation of processes at the top-level	1/47	2/41	0.71		0.48
6.10 wrong assignment of a process/activity into an object state	9/47	0/41	-2.96	OPM	0.003
6.11 missing process/activity	18/47	0/41	-4.44	OPM	<0.001
6.12 confusion between an activity and an action	4/47	0/41	-1.91	OPM	0.06
7.1 scaling rules incorrectly applied	11/47	10/41	0.11		0.91
8.1 missing events	22/47	6/41	-3.23	OPM	0.001
8.2 incorrect specification of an event	16/47	0/41	-4.13	OPM	<0.001
8.3 confusion between events and conditions	5/47	1/41	-1.52		0.13
8.4 incorrect specification of a condition	1/47	0/41	-0.94		0.35
8.5 incorrect specification of a temporal constraint	3/47	1/41	-0.89		0.38
9.1 missing states	3/47	0/41	-1.65		0.10
9.2 overlapping states	1/47	0/41	-0.94		0.35
9.3 missing transitions between states	16/47	1/41	-3.75	OPM	<0.001
9.4 assigning transitions between states which should not have been connected	8/47	1/41	-2.25	OPM	0.024
9.5 wrong assignment of states to objects	1/47	0/41	-0.94		0.35
9.6 irrelevant states	1/47	0/41	-0.94		0.35
10.1 no separation between real-world values and their measured values	1/47	7/41	1.98	OMT	0.05
10.2 a monitoring process updates the real-world value instead of the measured value	1/47	1/41	0.10		0.92
10.3 a monitoring process uses as input measured values instead of real-world values	1/47	0/41	-0.94		0.35

Table 4. Overall results of the Specification Quality part of the experiment.

		OMT		OPM		T-value	P-value	Difference in favor of
		Average	STDV	Average	STDV	two tailed		
Overall number of error types per student	all issues count the same	5.81	2.95	2.80	2.11	-5.36	0.01	OPM
	issues in favor of OMT count 4 times as much	8.49	4.69	6.32	4.60	-2.16	0.05	OPM

3.3 Participants Preference between the Two Methods

As was discussed in Section 2.2, we measured the student's satisfaction of using the two methods by their stated preference of the two methods. The preference was given according to a 7-point scale, ranging from absolute preference for OPM, being 1, to absolute preference for OMT, being 7. Averaging the scores given by the 65 (out of 88) students who answered the short questionnaire, we received an average score of 2.58 with standard error of $1.79/\sqrt{65}$, which is much less then the expected average of 4.

4. Discussion

In this section we discuss the experimental results and propose possible explanations to the significant differences found between the two methods in the Specification Comprehension and the Specification Quality experiment parts.

4.1. Specification Comprehension

The results show that there were significant differences between the two methods in specific issues. A significant difference in favor of OPM was found in three issues: (1) identifying triggering events of processes; (2) identifying processes that are triggered by a given event; and (3) identifying objects that participate in a process. A significant difference in favor of OMT was found in two others issues: (1) identifying events that affect objects by changing their state, and (2) determining process execution order.

The differences in favor of OPM, in the three issues listed above, can be explained on the basis of the different number of models of the two methodologies. Answering questions related to these issues in OPM entails locating a process in the single OPM model and following the links attached to that process. In OMT, identifying triggering events of processes and identifying processes triggered by an event requires consulting two OMT models: the Dynamic Model (Timed Statecharts) and the Functional Model (DFD). Identifying objects that participate in a process also requires two models: the Functional Model and the Object Model, the latter for identifying the object class to whom the process serves as a service.

As noted, there are two issues in which OMT has been found to be better than OPM: identifying events that affect objects by changing their state, and determining the relativity of process execution time. These advantages can also be explained on the basis of the number of required models. The information for both these issues can be retrieved from the Dynamic Model of OMT alone, without the need for any one of its two other models. In the single OPM model, a process separates each triggering event from the object whose state is changed, and this probably complicates the relation between the object's state change and the corresponding triggering event. Retrieving information from a single OMT model can thus be easier than retrieving the same information from the single OPM model, which is more complex. However, no significant differences between the two methods were found in the five other issues that involve just a single model of OMT. These issues are:

(1) Identifying objects that are affected by an event, which involves OMT's Dynamic Model;

(2) Identifying object states, which also involves OMT's Dynamic Model;

(3) Identifying structural relationships between objects, which involves OMT's Object Model

(4) Identifying cyclic processes, which can be spotted by the self-looping invocation link in the OPM model and in the self looping transitions in OMT's Dynamic Model of OMT;

(5) Identifying timing exceptions, which again involves only the Dynamic Model of OMT.

4.2. Specification Generation Quality

Overall, our findings show that the Specification Quality of the OPM group was significantly better than that of the OMT group. Note that this is *exactly* the same group that performed worse in the Specification Comprehension part, in which the student subjects used OMT. Looking carefully into these error types, we have classified them into the following groups: (a) Model multiplicity-related errors; (b) Graphic expressive power-related errors; (c) method specific errors; and (d) domain specific errors.

(1) Missing features, which involves OMT's Object and Functional models;

(2) Assigning processes/activities into object states, which involves OMT's Dynamic and Functional models;

(3) Missing activities, which involves OMT's Dynamic and Functional models;

(4) Missing transitions, which involves OMT's Dynamic and Functional models[1];

(5) Assigning transitions between states which should not have been connected, which involves OMT's Dynamic and Functional models[1].

The fact that students who used OPM made fewer errors of the above types can be explained by the use of a single model in OPM, as opposed to the multiple models of OMT. Since generating a single-model OPM specification involves no transition between models, no errors in which some information is erroneously omitted from one of the models, or presented in one of the models in a way that is inconsistent with the way it is presented in another model can occur.

Graphic expressive power related errors

The error types in this category consist of incorrect specification of events and distinguishing between parallel and mutually exclusive processes. Both can be explained by differences in the graphic expressive power between the two methods. In OPM, specifying events relies mostly on graphically linking the triggering object to the triggered process rather than on text. Distinguishing between parallel and mutually exclusive processes is done graphically both in OMT's Timed Statecharts and in OPM. It seems that the graphic notation used in OPM was clearer to the students.

Method-specific errors

Some errors can only be made in one method but not in the other. Thus, confusing a datastore with an external entity, and confusion between an activity and an action can occur only in OMT, while errors of confusing an instrument

[1] Although state transitions are modeled only in the Dynamic Model, some state changes result from process execution. Therefore, they should be considered while the Functional Model is composed.

link with an effect link can occur only in OPM. Significant differences between the methods in these error types are therefore obvious.

Domain-specific errors

Domain specific error types, in particular the error of not distinguishing real-world values from measured ones, occurred more in OPM. This may be due to the Functional Model of OMT, DFD, which forces the analyst to distinguish between external (real world) entities and data-stores (representations or real-world entities in the model).[2]

4.3. Critical Review of the Experiment Setting

One possible criticism on the experiment described in this research may be that the authors, who developed OPM and OPM/T, were the same people who taught and tested it, a fact that might have affected the students and the experiment outcomes. Another possible criticism is that the fact that OMT was taught first might have a positive learning effect on the learning of OPM. While these may seem like critical factors, we made it a point to teach both methods as alternatives without expressing explicit opinions about the superiority of one method over the other. Moreover, OMT was taught for a longer time (total of 6 weeks vs. 5 for OPM) and the midterm examination which the students took included only OMT, so overall they spent more time on studying OMT than OPM. The students did their best in the experiment since the test determined 70% of their final grade. It is important to emphasize that due to the crossover experimental design, both groups of students performed better while working with OPM (in either specification or comprehension parts). This eliminates any potential capability difference between the two groups that may exist despite of the random assignment of the students into the two groups.

In the Specification Quality part of the experiment, we measured the overall quality of the specifications by the mean number of error types per student. We did not distinguish between more important and less important errors. It will be interesting to develop a grading scheme that assigns a proper weight for each error type.

In the Specification Quality part of the experiment, we instructed the students to use zooming/scaling in their solutions. There was no significant difference between OPM and OMT in the number of errors due to incorrect application of scaling rules. Although the problem that the students were asked to specify was not too large in size, OPM solutions require the use of the scaling mechanism in order to avoid the problem of diagram cluttering that results from the fact that all system aspects are shown in the single OPM model. All students were asked to use scaling, the ones using OMT as well as OPM, so that their specifications will be equivalent. In this experiment, we did not examine the design choices made by the students as to which of the processes were to be zoomed into. Further experimentation, including a larger scale problem, could examine this issue.

Summary

This study reports the results of a double-blind controlled experiment designed to compare the expressive power of OPM and OMT. The comparison was done in terms of (a) the quality of the resulting system specification, as measured by the nature and quantity of specification error types; and (b) specification comprehension, i.e., how easy it is for users to interpret and understand a given specification. To obtain conclusive results, real-time systems, which exhibit a more complex dynamic behavior than non-real-time systems, were selected as the focus of the experiment.

---

[2] At the time of the experiment OPM did not have the facility to distinguish between system and environment objects and processes (the latter marked with dashed contours), and between information and physical objects and processes (the latter marked with shadowed contours).

The experiment outcomes and their rigorous statistical analysis establish that Specification Quality is significantly better in OPM than in OMT. This difference is due to specific issues, in which significant differences between the two methodologies were found. The Specification Comprehension results show that there were significant differences between the two methods in specific issues.

Our conclusions are based on a specific experimental setting, i.e., certain tasks, subjects and analysis methods. The tasks were not trivial in terms of size and complexity, the subjects were not novices, and the analysis methods were not trivial.

The major difference between OPM and OMT is the singularity of the Object-Process model in OPM, as opposed to the multiplicity of models in OMT. As we have shown, the source of most of the errors in OMT is the lack of integration among the method's different models and the need to maintain consistency and gather information that is scattered across these models. This is consistent with the theory that when the specification is done by a combination of different models, an understanding of the system's structure and behavior is achieved by a mental integration of the different models. The burden of this mental integration, which is put on the shoulders of the analyst and his/her target audience, is not an easy task. Moreover, incompatibility problems, such as mismatches among names of objects and processes, are more likely to occur when more than one model is used.

This model multiplicity problem is inherent in any systems development method that uses a number of models to obtain a complete system specification. OPM and its temporal extension, OPM/T, avoid this problem in the first place by incorporating all the information within a single model and managing the complexity through an elaborate scaling mechanism. Perhaps the most significant conclusion of this study is that *the likelihood of making errors increases with the number of models in the method.*

The students who took part in the experiment significantly preferred OPM to OMT.

The efficiency (the time it takes to complete the task) of specification comprehension and specification generation was not a measure in this experiment, due to the fact that the subjects of the experiment knew that their grade depended only on the effectiveness of their solutions, and on their efficiency.

There is room for further experimentation and comparison of the two methods. Such experiments should be conducted from the point of view of both users and designers, and should be based on larger and more complex systems. They should also examine and measure the time and effort needed for learning the method and using it for modeling complex systems.

It is our conjecture that OPM will exhibit similar advantages with respect to other recognized object-oriented methods at least to the same extent as it did with respect to OMT. We base this conjecture on the fact that the more recently developed object-oriented methods feature more models. In particular, UML [11] includes nine different diagram types. This exacerbates the model multiplicity problem, since the number of model interactions increases exponentially with the number of models in the method.

References

1. Y. Wand and R.Y. Wang, "Anchoring Data Quality Dimensions in Ontological Foundations", *Communications of the* ACM, Vol. 39, No. 11, November 1996, pp. 86-95.
2. M. Peleg, and D. Dori, "Extending the Object-Process Methodology to Handle Real-Time Systems", *Journal of Object-Oriented Programming*, Vol. 11, No. 8, January 1999, pp.53-58.

3. D. Dori, "Object-Process Analysis: Maintaining the Balance Between System Structure and Behavior", *Journal of Logic and Computation,* Vol. 5, No. 2, 1995, pp.227-249.
4. J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorenson, *Object-Oriented Modeling and Design* Prentice-Hall Englewood Cliffs NJ, 1991.
5. Y. Kesten, and A. Pnueli, "Timed and Hybrid Statecharts and their Textual Representation", *Lecture Notes in Computer Science* Vol. 571 Springer-Verlag, 1991, pp. 591-620..
6. D. Drusinsky, and D. Harel, "Using Statecharts for Hardware Description and Synthesis", *IEEE Transactions on Computer-Aided Design*, Vol. 8 , No. 7, 1989.
7. H. Toetenel, and J.V. Katwijik, "Stepwise Development of Model-Oriented Real-Time Specifications From Action/Event Models", *Lecture Notes in Computer Science,* Vol. 571 Springer-Verlag, 1991, pp. 547-570.
8. P.H. Laplante, *Real-Time Systems Design and Analysis: An Engineer's Handbook* IEEE Society Press, NY, 1993.
9. P. Shoval, and I. Frumermann, "OO and EER Conceptual Schemas: A comparison of User Comprehension", *Journal of Database Management* Vol. 5, No. 4, 1994, pp. 28-38.
10. Siegel, S. and Castellan, N.J., *Nonparametric Statistics for the Behavioral Sciences* McGraw-Hill NY, 1988.
11. G. Booch, I. Jacobson, and J. Rumbaugh, *Unified Modeling Language (UML) Notation Guide Version 1.1*, Rational Software Corporation, September 1 1997.

Appendix A: The Case Studies Used in the Experiment

In this Appendix we present the formal problem statements for the Specification Comprehension and Specification Quality case studies used in the experiment.

1. The controller starts monitoring the room temperature within 1 second after the main switch is turned to the heat position.
2. The controller sends a signal to activate the motor 1 second after the monitored room temperature has fallen below (H_d-2) degrees, where H_d is the desired temperature.
3. The controller monitors the motor speed. Within 1 second after the speed has reached a predefined value, S_d, the controller signals the oil valve to open and ignites the furnace.
4. The controller also monitors the water temperature. Within 1 second after the temperature has reached a predefined value, T_w, the controller signals the circulation valve to open. The heated water then starts to circulate through the house and heat it.
5. A fuel level indicator and a combustion sensor send exception signals to the controller if abnormalities occur, in which case the controller shuts the furnace off within 0.1 seconds.
6. Once the home temperature reaches (H_d+2) degrees, the controller deactivates the furnace first by closing the oil valve within 1 second, and 5 seconds later, by closing the circulation valve and stopping the motor.

In addition to the specifications, the system must also abide by the following constraints:

- The minimum time interval between the Furnace Motor restart and the end of the most recent operation is 5 minutes (300 seconds).
- The Furnace cannot run for more than *max* consecutive seconds.

Figure A-1: The Home Heating System Problem Statement

An avionics navigation system uses an accelerometer to monitor the aircraft's engine acceleration every 5 milliseconds and a gyroscope to monitor its angle every 40 seconds. Each time the measured aircraft's angle is updated, its navigation system uses this data, the measured acceleration, and the previously calculated values of the aircraft's velocity, height and position, in order to calculate the new velocity, height and position of the aircraft. Assume that there is always available data concerning the velocity, height and position of the aircraft, and that they are initialized according to the aircraft's take-off position.

The aircraft velocity, height and position precise calculation process takes no less than 40 milliseconds and should take no more than 100 milliseconds. If the process exceeds this limit, an alternative calculation process takes place. It receives as input the measured engine's acceleration, the measured aircraft angle and the most recently calculated aircraft velocity, height and position. As soon as the aircraft velocity, position and height are established, the navigation system controls the aircraft's flaps according to the aircraft'