A controlled experiment investigation of Behavior allocation in two Object-Oriented methods

Ignatios Deligiannis[1], Panagiotis Sfetsos[2], Alexander Chatzigeorgiou[3]

[1,2] Dpt. of Information Technology, Alexander Technological Education Institute of Thessaloniki, Greece
Tel: +30 2310791295, E-mail: ignatios, sfetsos@it.teithe.gr

[3]Dpt. of Applied Informatics, University of Macedonia, Thessaloniki, Greece,
Tel: +30 2310891886, E-mail: achat@uom.gr

This paper presents a controlled experiment study, investigating the impact of design techniques on a central and critical activity in designing Object-Oriented systems, *behaviour allocation,* implemented by assigning responsibilities to classes. Two alternative designing methods were used; Rational Unified Process (RUP) versus ICONIX method. Two groups of undergraduate students participated, each performing on one of the two designing methods. The subjects designed, using UML, a given use-case and a domain model, in order to assess the degree of their effectiveness in assigning methods in classes, examined by three quality factors: *completeness, correctness* and *consistency*. The results indicate that concerning *effort, completeness,* and *consistency* participants in RUP method performed more effectively than those performing in ICONIX method.

Keywords

Object-Oriented, controlled experiment, design method, quality factors.

1. Introduction

Assigning responsibilities to classes is a central and critical activity in designing Object-Oriented (OO) systems. This concerns the allocation of behavior in classes, the dynamic dimension of a system. Namely, first to define a method and then decide where, in which class, to allocate it. It is an activity where the designers have to choose between alternative solutions aiming at building a well designed, robust, understandable and maintainable system.

The design of an OO system involves two main kinds of models, one dealing with the static structure, and the other dealing with dynamic behaviour. Domain model and class diagram are considered to belong in the *static model*, since they capture the static structure of a system, like entities, attributes and their types, and relationships with their multiplicities. The *dynamic model* is actually capturing the behaviour of the system, i.e. how the objects react and communicate to each other in order to fulfill the system requirements. The activity of behavior allocation, determinant for the systems functional structure and quality, is considered the most difficult in practice, due to difficulties practitioners face firstly in method definition and the secondly in their allocation in classes [1, 10]. Aiming at easily and effectively accomplishing this task, a few proper techniques have been developed within different OO methods. Such techniques emphasizing the assignment of responsibilities in classes are: Class-Responsibility-Collaboration (CRC) cards [2], General-Responsibility-Assignment-Patterns (GRASP) [3], and Robustness Analysis [1]. Rational Unified Process (RUP), using GRASP patterns [3], and ICONIX, using Robustness Analysis [1], have gained a broad acceptance within industry, therefore are more challenging for empirical evaluation.

In this study we focus on building of Sequence Diagram implemented by the two methods. Both methods adopting a different approach, as described above, allege that aim at effectively building robust and quality detailed designs, seen from the designer perspective. In order to compare the effectiveness of the two methods in responsibility assignment, we designed and conducted a controlled experiment, with undergraduate students as participants. The reason we choose Sequence diagram than collaboration diagram is that it is considered more understandable [4].

The paper is structured as follows. Section 2 describes the details of the experiment. Section 3 summarizes the data collected and presents the data analysis results. Section 4 identifies and discusses possible threats to the validity of the study. Finally, section 5, presents the conclusions and future search aims.

2. Description of the experiment

The present study is considered to be a controlled experiment, and key motivator for using formal experimentation is that the results can be more easily generalised than those of an observational or a case study [7]. It concentrates on the investigation of behavior allocation through method assignment in Sequence Diagrams. Our decision to concentrate on the specific designing aspect stems from the considered difficulty practitioner face in performing on this activity, as well as, the existence of different techniques insisting to effectively contributing on producing robust and quality design structures.

2.1 Responsibility assignment

The sequence of activities from Use case, which captures the requirements specification, to Sequence diagram, capturing method allocation, for both methods (RUP, ICONIX) has as shown in figure 1. Three of them (use case, domain model, and sequence diagram) are common. Within this set of activities, the two methods mainly differentiate in that of responsibility assignment. Specifically, RUP starts with the System Sequence Diagram (SSD), defining the use case course of actions (events) to the system, seen the system as a black-box. Then, refolding each action, tries to find out the subsequent sequence of methods – messages exchanging between classes – in a Sequence Diagram, in order to fulfill the needed behavior. This process is guided by a number of proper guidelines called GRASP patterns [3].

ICONIX on the other side, firstly tries to clarify the use case behavior into the Robustness diagram. There, the sequence of user actions on the system and the subsequent sequence of functionality is unfolding step by step, designed by a few cooperating symbols: boundary object, control object and entity object. Following is a Preliminary Design Review, a review activity where each scenario from the use case is inspected in order to ensure that previously created artifacts smoothly match, capturing the desired system behavior [1]. Then, follows the Sequence Diagram implementing in methods the behavior allocation.

The responsibility assignment activity is implemented by the two methods as depicted in figure 1.

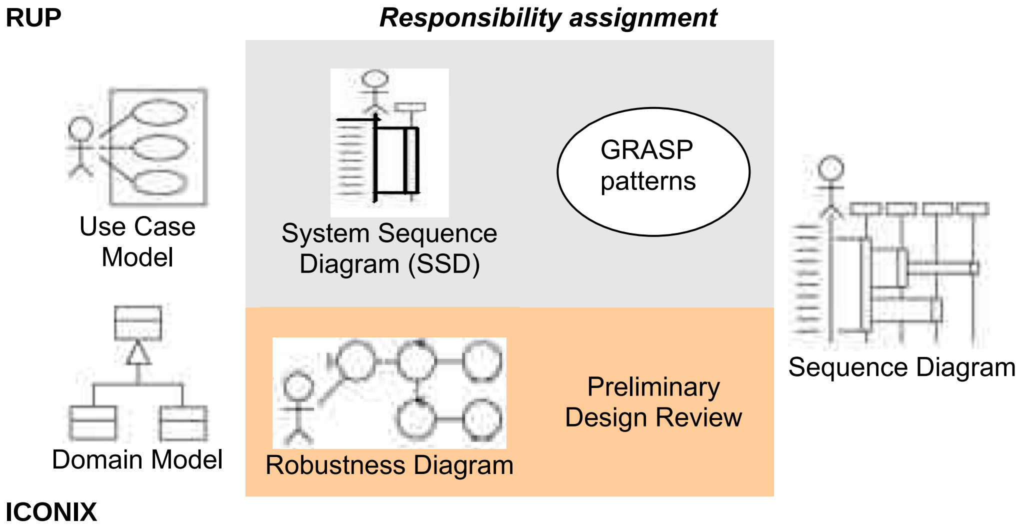

Figure 1. Activities of the two methods

2.1. Hypotheses

For hypotheses testing the deliverables of design artefacts by the participants documents of two functionally equivalent design versions of a system, called Hi-Fi management system (HFMS), were used. In order to investigate the *effectiveness* of two OO methods in behaviour assignment in classes, *standard significance testing* was used, the null hypothesis being stated as:

H_0 There is no difference between the two OO methods in assigning responsibilities, in terms of effectiveness assessed by effort, completeness, correctness and consistency.

The alternative hypotheses, i.e., what was expected to occur, were then stated as:

The RUP process is more effective, in assigning responsibilities than ICONIX process in terms of:

- H_1 effort spent (H_{1a} total, H_{1b} responsibility assignment).
- H_2 completeness.
- H_3 correctness.
- H_4 consistency.

2.3 Experimental material - Design

Thirty two second year students of the Dept. of Informatics at the Alexander Technological Education Institute, Thessaloniki, Greece, voluntarily participated. They were taught the RUP and GRASP patterns, as those suggested by C. Larman [3], as well as the ICONIX process [1]. The students were randomly allocated in two experimental groups, 18 in R-group, performing on R-design (RUP) and 14 in I-group, performing on I-design (ICONIX), each named according to the first letter of the corresponding method. They had to perform under similar conditions on a given task. Namely, each group was given the same use case specification and the same domain model, aiming each to produce two artifacts: a System Sequence Diagram and a Sequence Diagram, for the R-group, and the Robustness Diagram and the Sequence Diagram, for the I-group. However, from I-group one participant spent a lot time in creating the Robustness diagram, thus he did not complete the Sequence diagram on time, so his deliverables were not considered as properly finished and dropped from our study. In the rest of the paper we call them as R-group and I-group for convenience, according to the system they worked on.

The application domain used for the system was a small but realistic application, implementing the use case “make a repair transaction” of HiFi device/s. The given documentation consisted of: 1) the use case specification, containing 19 steps (sentences), 2) proper prototypes, portraying the user interface, and 3) the domain model, consisting of 7 conceptual classes. 4) R-group was additionally given a set of nine GRASP patterns. Both groups were expected to deliver a detailed Sequence Diagram, taking care for quality factors, as: completeness, correctness and consistency [5].

There were two tasks to be performed by the participants. First, they were given the documentation set in order to understand the required functionality. The task was designed to match as much as possible a real situation. Namely, a customer could bring to the store one or more HiFi device/s requesting for repair. The system would check for each device’s purchasing date, guarantee, etc., performing a preliminary estimation for the repair cost and time required. Thus, the above mentioned artifacts were expected from each participant along with additional information explaining his/her performance time. The second task required to complete a debriefing questionnaire, which captured, i) their personal details and experience, ii) their understanding of the system, and iii) time spent (effort) in required artifacts.

Procedure. A preliminary test was conducted in order to verify the capability of the volunteers to participate in the experiment. The test questions were related to various design concepts, e.g., domain modeling, message passing, etc., that were applied within the experiment designs. Also, a pilot run was performed using two students, in order to test the experimental settings before the experiment proper took place. The experiment proper was performed in a classroom, where the participants were randomly placed in the working places. Extra care was taken to eliminate plagiarism, although it was not a real concern.

The participants were told verbally that there were different documents being worked upon, but they were not told anything about the nature of the study, i.e., what hypotheses were being tested. They were told to complete their tasks correctly, exploiting efficiently their time. Completing each artifact or activity they had to mark the current time. They were given a maximum of two hours to complete the tasks. They were also told that any questions could be directed towards the two experiment monitors. Nevertheless, questions were not answered if thought to assist participants' performance. Only explanations were provided to them, avoiding affecting their performance. After completing their tasks they called one of the experiment monitors to collect their documentation. This was necessary in order to mark their performance time as well as avoiding any further changing. Finally, they were given a debriefing questionnaire to complete, expressing their subjective opinion concerning a number of issues as described above.

Design. In this study the type of design applied is a completely randomized balanced design [7]. The independent variable is behavior assignment in OO methods under investigation. The experiment context is characterized as multi-test within object study [7], p. 43). An advantage of using such a design is that it simplifies the statistical analysis.

2.4 Dependent variables

In order to test the hypotheses examining the effectiveness of the two OO methods, quantitative and qualitative analysis is required. Objective criteria depend upon data gathered through measurement. According to Whitmire's [6] measurement framework, this study is most closely related to a *technical point of view*, and to objects *products* and *processes*. From the processes the external attribute *effort* is measured, while from the products the internal attributes captured by the three quality factors, *completeness*, *correctness*, and *consistency,* are measured.

Thus, **Effectiveness** is assessed by effort and the three quality factors for OO design quality assessment. All of them provide objective measurements. The assessment process was based on the criteria applied on each of the nineteen specific sentences consisting the use case specification, for both groups' deliverables.

3. Experimental results

3.1 Statistical analysis of the data

For the statistical analysis of the data we used variables related to participants' performance in the experiment. The statistical methods we used to test the hypotheses were: the Student's *t*-test for independent samples, in order to test significant differences between the two groups (R-group and I-group) for each individual variable. Significance level was set to $\alpha < 0,1$. Hence, each hypothesis was examined by the corresponding variable, as described following:

Effort, was examined by (hypotheses H_{1a} and H_{1b}) *time spent* by the participants to perform on the required activities for each method (table 1), in two ways: a) *total effort,* H_{1a}*,* counting total time spent from studying the given Use case to designing the Sequence diagram, where there is evidence for supporting H_{1a} hypothesis; b) only counting time spent for responsibility assignment activities, where there is no evidence for supporting H_{1b} hypothesis (table 2).

Table 1. Activities required by the two methods from Use case to Sequence diagram

Activity	RUP	ICONIX	Effort	
			Total	**Responsibility assignment**
Use case study	✓	✓	✓	
Requirements Review		✓	✓	
System Sequence Diagram	✓		✓	✓
Robustness analysis		✓	✓	✓
Preliminary Design Review		✓	✓	✓
Sequence Diagram	✓	✓	✓	✓

For the three quality factors, we consider worth mentioning the procedure we followed to evaluate the above stated criteria. Namely, we applied those on each specific sentence of the use case scenarios, since each sentence describe a specific task as well as functionality. Therefore, results extracted from the statistical analysis are shown in table 2.

Table 2. Hypotheses testing

Hypotheses	*p*-value	*α<0,1*
$H_{1a,}$ *total effort*	0,000	support
$H_{1b,}$ *effort for responsibility assignment*	0,110	
$H_{2,}$ *completeness*	**0,096**	support
$H_{3,}$ *correctness*	0,776	
H4, consistency	**0,025**	support

4. Threats to validity

This section discusses various threats to validity and the way we attempted to alleviate them.

Considering the ***construct validity***, i.e. the degree to which various factors accurately measure the concepts they purport to measure, the following possible threats have been identified: (a) Effort spent by the participants, was measured by time spent, which is considered an objective measure. The fact that a participant could not complete in time, does not affect the results. The quality of the delivered solutions (artifacts) was examined objectively by *three* criteria (completeness, correctness, consistency).

Threats to ***internal validity*** are influences that can affect the independent variable with respect to causality. The following possible treats are identified:

(a) *Instrumentation*. This is the effect caused by the differences in the artifacts used. We consider it is of no concern, since both groups performed on the same task, both designs were performing identically. (b) *Selection.* This is the effect of natural variation in human performance.

Volunteer students were used, considered more motivated and suited for the task than the whole population.

Regarding the ***external validity***, i.e. the degree to which the results of the research can be generalized to the population under study and other research settings, the following possible threats have been identified:

(a) This study took place in a university environment and not in the work place. (b) The size of the system used although relatively small was a real project, and the task essentially was a design creation, that occurs in practice in small systems too. (c) The students may not be representative of OO software professionals. However, Briand *et al.,* [8] argue that student based experiments can provide useful results for identifying interesting hypotheses that are worthy of further investigation in more realistic settings.

5. Conclusions

This study has investigated the effectiveness of two OO methods on behaviour assignment in classes.

The obtained results do not support the effectiveness hypothesis. However, they provide sufficient evidence that the specific design techniques can affect the effectiveness in designing OO systems. This is supported by the analysis results for the quality factor of completeness and consistency.

Examining each dependent variable separately, the findings indicate that concerning *total effort*, RUP method is more effective than ICONIX, although it is not, concerning the time spent for responsibility assignment that we are most interested. An interpretation for the total effort could be that RUP is more straightforward, including two activities, while ICONIX includes four (table 1), including two reviews, where considerable time was spent by the participants.

Concerning *completeness*, supported by the hypothesis H_2, the fact that ICONIX giving extra emphasis to traceability, is forced to some repetition of design members. Our opinion based on the examination of the deliverables, is that participants attempting to transform the control objects into methods (operations) into Sequence diagram, face an addition difficulty, resulting in loose of design members. Similarly, this effect could possibly affect the *consistency* variable, supported by H_4.

Considering the dependent variable *correctness*, not supported by hypothesis H_3, our explanation is that ICONIX, aiming at traceability leads in more correct designs.

Future research, planned as a result of this study we consider would focus on the issue of the quality of interaction modelling, that has not been adequately addressed yet [9]. Namely, to build a model for Sequence diagrams, which should: a) emphasize on traceability, originating from ICONIX; b) guided by design heuristics, in order to improve the designers performance on the above examined quality attributes[11]; c) as well as to contribute in building quality design structures, that could easily be assessed by proper measures, found in bibliography [12].

Acknowledgements

This work is funded by the Greek Ministry of Education (25%) and European Union (75%) under the EPEAEK II program “Archimedes II”.

References

[1] D. Rosenberg, M. Collins-Cope, M. Stephens, “Agile Development with ICONIX Process: People, Process, and Pragmatism”, Apress,

[2] S. W. Ambler, “The Object Primer: Agile Model-Driven Development with UML 2.0”, *Cambridge University Press*, 2004.

[3] C. Larman, “Applying UML and Patterns”, Prentice Hall, 2002.

[4] M. C. Otero, J.J Dolado, “An Initial Experimental Assessment of the Dynamic Modelling in UML”, *Empirical Software Engineering*, Vol. 7, no 1, pp. 27-47, 2002.

[5] M. L. Major, J. D. Mcgregor, “Using Guided Inspection to Validate UML Models”, *http://sel.gsfc.nasa.gov/website/sew/1999/topics/major_SEW99paper.pdf*

[6] Whitmire, S., Object Oriented Design Measurement, *John Wiley & sons*, 1997

[7] C. Wohlin, et. al., “Experimentation in Software Engineering - An introduction”, *Kluwer Academic Publishers,* 2000.

[8] L. Briand, C. Bunse, and J. Daly, A Controlled Experiment for Evaluating Quality Guidelines on The Maintainability of Object-Oriented Designs. *IEEE Trans. on Softw. Eng.*, Vol. 27, no 6. p. 513-530, 2001.

[9] C. Glezer, M. Last, E. Nahmani , P. Shoval, “Experimental Comparison of Sequence and Collaboration Diagrams in Different Application Domains”, *Conf. Tenth CAISE/ IFIP8.1 Int'l Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD’05*), 2005.

[10] T. Sin, “Improving Usability of Analysis Sequence Diagram in Transaction-Oriented Applications", *Proc. AMCIS’07*, 2007.

[11] Il-Yeol Song, “Developing Sequence Diagrams in UML” *LNCS*, Vol. 2224, pp. 368-382, 2001.

[12] Diagrams, A. Rountev, S. Kagan, J. Sawin, ”Coverage Criteria for Testing of Object, Interactions in Sequence”, LNCS 3442, pp. 289–304, 2005.