

Impact of Domain Modeling Techniques on the Quality of Domain Model: An Experiment

Hiqmat Nisa

Dept. of Computer Science & Software Engineering
International Islamic University
Islamabad, Pakistan

Muhammad Uzair Khan

Dept. of Computer Science National University of Computer
& Emerging Sciences (FAST-NU)
Islamabad, Pakistan

Salma Imtiaz

Dept. of Computer Science & Software Engineering
International Islamic University
Islamabad, Pakistan

Saima Imtiaz

Dept. of Computer Science & Software Engineering
International Islamic University
Islamabad, Pakistan

Abstract—The unified modeling language (UML) is widely used to analyze and design different software development artifacts in an object oriented development. Domain model is a significant artifact that models the problem domain and visually represents real world objects and relationships among them. It facilitates the comprehension process by identifying the vocabulary and key concepts of the business world. Category list technique identifies concepts and associations with the help of pre defined categories, which are important to business information systems. Whereas noun phrasing technique performs grammatical analysis of use case description to recognize concepts and associations. Both of these techniques are used for the construction of domain model, however, no empirical evidence exists that evaluates the quality of the resultant domain model constructed via these two basic techniques. A controlled experiment was performed to investigate the impact of category list and noun phrasing technique on quality of the domain model. The constructed domain model is evaluated for completeness, correctness and effort required for its design. The obtained results show that category list technique is better than noun phrasing technique for the identification of concepts as it avoids generating unnecessary elements i.e. extra concepts, associations and attributes in the domain model. The noun phrasing technique produces a comprehensive domain model and requires less effort as compared to category list. There is no statistically significant difference between both techniques in case of correctness.

Keywords—Domain Model; UML; Experiment; Noun Phrasing Technique; Category List Technique

I. INTRODUCTION

UML (Unified modeling language) is gaining fame since its inception in 1997; it is being commonly practiced by the industry to model object oriented software systems. UML plays a significant role in reducing the complexity of large software system by modeling different aspects throughout SDLC phases. Object Oriented Analysis (OOA) is carried to understand and model the problem domain in the form of real world objects, which can later be translated into the solution. It describes problem domain from the perspective of objects and emphasizes on identifying and describing the concepts,

attributes and associations in the problem domain [1]. One of the main outcomes of OOA is a domain model which models the problem domain objects along with their associations and attributes.

Domain model is one of the most important UML artifact used to understand the problem domain. It represents vocabulary and key concepts, important to the business world [1] [2] [3] and consists of visual representation of concepts, attributes and association among conceptual classes in the real world domain. It also presents general vocabulary, which helps in clear communication between the team members and helps elevate the level of understanding between the development team and customer side [2] [3]. A solution which is representative of the customer needs requires a domain model that is representative of the domain. A clear and precise domain model can also help in reducing risk [4] and effort and cost of rework required at later stages [5]. Therefore one of the major goals of OOA is to create an accurate and complete domain model.

The domain model can be created using two different techniques suggested by Larman [1]: category list technique and noun phrasing technique. To identify potential candidate classes and associations; category list technique provides a list of categories which are usually important to business information systems. Each category represents entities or concepts related to real-world. Sets of candidate classes produced by all categories are quite independent from each other whereas, noun phrasing technique is linguistic analysis. Noun phrasing technique involves the identification of nouns and noun phrases in the domain description, and considers them as conceptual classes or attributes [1]. These techniques have not been empirically evaluated for their effectiveness in creating a quality domain model. Therefore an experiment was performed to evaluate the effectiveness of techniques in creating a complete and accurate domain model. The experiment was conducted with help of undergraduate students of fourth semester of software engineering, as they are assumed to be familiar with the models and notations of UML. This experiment is focused to answer the below given research questions.

RQ1: What is the effect of noun phrasing and category list technique on the quality of the domain model?

RQ2: What is the amount of effort required to create the domain model using both techniques?

The quality of the domain model is determined on the basis of completeness and correctness of the domain model, whereas the amount of effort is measured in terms time taken to create the model. The rest of the paper is organized as follows: Section 2 presents the Background and Related Work. In Section 3 elaborates on the Design of the Experiment and Section 4 discusses the Analysis and Results. Finally conclusion and future work is given in section 5.

II. BACKGROUND AND RELATED WORK

Domain model is the most important and common model in object oriented analysis. It describes the noteworthy concepts or objects in problem domain. It is a representation of the real-world conceptual classes, attributes of the classes and associations among them Domain model is an improved version of the project dictionary, where the terms used in the project are present along with the graphical visualization of the connections between them. It can be termed as a simplified version of a class diagram, one that does not incorporate responsibility assignment [1]. Most of the conceptual classes modeled in domain model become part of the class diagram, which are important to software development [2] [6].

Domain model can be created using two different techniques namely: noun phrasing technique and category list technique [1]. There are some basic steps involved to create a domain model i.e. identification of conceptual classes along with their attributes and associations and unnecessary candidate classes.

Noun phrasing technique uses grammatical analysis of use case description to identify nouns and noun phrases and consider them as candidate conceptual classes or attributes. For the identification of associations, verb phrases are identified between entities and are considered as relationships between conceptual classes. However, for the identification of potential candidate classes and associations using category list technique Larman [1] provides a list of categories which are usually important to business information system and also provides guideline to eliminate useless concepts which are not appropriate to be implemented

Noun phrasing technique is the simplest approach to create domain model, but result in many imprecision problems e.g. words may be ambiguous or the identification of redundant classes due to synonyms in use case description and noun phrase may also be an attributes rather than a concept [1]. Identifying noun and noun phrases is an analyst's job to examine each noun phrase and consider it either as a concept or an attribute. Some guidelines have been proposed by Larman to identify and refine attributes. The research focuses on empirically evaluating both of the techniques to observe their effect on the quality of domain model.

The Literature survey highlights that various empirical studies have been conducted to evaluate the impact of different techniques used to construct different UML models. Most of the target UML models are use case diagram, Class diagram and sequence diagram. The work of T. Yue et.al. [7] for instance, investigated whether restricted use case modeling (RUCM) approach or traditional use case template produced high quality analysis models i.e. Class diagram and sequence model. Subjects designed a class and sequence diagram of a given software systems using RUCM approach and traditional use case template. Results pointed out that RUCM produced better quality model than traditional use case template. Similar experiment was performed S.Tiwari *et al.* [8] [9], where they investigated the impact of use case templates on the quality of class diagram and use case diagram. They concluded [9] that no template is statistically significant better over another in terms of completeness, consistency, understandability, redundancy and fault proneness. However formal use case template produced high quality class diagram as compared to UML use case template and formal use case produced less redundant elements in class diagram [10]. Another study I [11], evaluated the effectiveness of two techniques i.e. validation and derivation technique on the quality of class diagram, and concluded that derivation technique produced more complete class diagram as compared to validation technique.

The quality of domain model is also evaluated by some researchers. The impact of system sequence diagram (SSD) and system operation contract (SOC) is observed on the quality of domain model [12]. The subjects designed domain model with SSD and SOC and without SSD and SOC. Two factors were involved to evaluate the quality of domain model, i.e., completeness and time. Author concluded that using SSD and SOC to construct a domain model, improves the quality of domain model in case when subjects have enough practice to take advantage from SSD and SOC. Another study conducted by S. Espana et al. [10] evaluated the quality of conceptual model constructed by two alternative techniques i.e. text-based derivation technique and communication based derivation technique. Participants were required to construct conceptual model using two alternative techniques. The quality of derived conceptual model was evaluated according to completeness and number of faults present (model validity) in participants conceptual model. The results highlight that the participants who used communication based derivation techniques produced 9.22% more complete conceptual diagram as compared to those who used text based derivation technique. Briand et al. [12] investigated that whether the use of SSD or SOC in domain model construction, improve the quality of domain model or not. Whereas, a main concern is to evaluate domain model construction technique suggested by Larman [1].

Most of the researchers conducted empirically studies, to compare different techniques [10] [7] [11] [13] for the purpose that which technique leads to high quality UML diagram. However most of the target UML models are class, use case diagram and sequence diagram. This research is focused on

the quality of resultant domain model created via noun phrasing and category list technique.

III. EXPERIMENT PLANNING

The research is validated with help of an experiment. This section explains design of the experiment. The experimental guidelines were followed to design the experiment in a controlled environment as suggested by C. Wohlin [14]. All the steps of an experiment to evaluate the quality of domain modeling techniques are reported in this section.

A. Experiment Definition

The purpose of this research is to empirically evaluate the impact of noun phrasing and category list technique on the quality of domain model. Our main concern is the creation of a domain model by the subjects via noun phrase or category list technique. As a result, two treatments are described as independent variable. One describes the creating of domain model using noun phrasing technique, and the other one describe the domain model using category list technique. The aim of this experiment is to evaluate the quality of domain model in terms of correctness, completeness, and effort required to design a complete domain model.

B. Context selection and subject

The selection of the subjects is very important for generalizing the results of experiment. Results generalization can be achieved by satisfactory sample size and random subject selection [14]. This experiment is conducted with 68 fourth year undergraduate computer science students in a famous Science and technology University of Islamabad, Pakistan. The students are familiar with UML notation and domain modeling techniques. They studied UML as part of their software engineering course in initial semesters. All the students have similar experience in modeling UML diagrams. The students were selected as experiment subjects as they fulfill the criteria i.e. participants who have similar education background, adequate knowledge and training of domain modeling.

To avoid biasness simple random sampling [14] is used for subject selection, i.e. subjects are selected from the population at random. Subjects were divided into two groups: group A and group B according to their grades. The categorization of students in two groups according to their grades is done to minimize the impact of students' capability on experiment's results. Before conducting the experiment a brief presentation is given to students about domain modeling techniques and the experiment. However the hypothesis of the experiment is not disclosed.

Two different systems were used as objects in this experiment, Automatic Teller Machine (ATM) and internet book store system (IBS). The ATM use case describes the process of withdraw fund and card verification as discussed in [15]. The IBS system purchases books over internet via credit card and Amazon website as discussed in [16]. We provide the experimental systems of limited complexity due to time constraints, so that subjects are able to finish their task.

C. Dependent and independent variable

There are two independent variables, Technique (category list and noun phrase) and Domain used (ATM and IBS).

Quality of domain model is evaluated by three dependent variables i.e. completeness, correctness and effort. Correctness is calculated in terms of average value of Useless Concepts(UC), Missing Concepts (MC), Extra Relationships (ER), Missing Relationships (MR), Extra Attributes(EA), Missing Attributes (MA) and Missing Generalizations(MG [12]. Completeness is defined as average of correctly identified elements in the domain model i.e. average number of Correct Concepts (CC), Correct Relationships (CR) and correct attributes (CA) and Correct Generalizations (CG) [7]. Table I and table II present the completeness of domain model completeness.

The second dependent variable checks the significant difference between the effort required to design a domain model by subjects who use noun phrase technique and those who use category list technique. The effort is calculated in terms of time, measured in minutes. Only that time was considered which utilized in creation of fully completed or partially completed domain model. The time is computed by subtracting the start time of the experimental task from end time of the experimental task.

D. Hypothesis

Two main research questions are investigated in this experiment. The first question contains a number of hypotheses shown in table III. According to experimental design one independent variable was considered called method, with two treatments: category list technique and noun phrasing technique, and three dependent variables correctness, completeness of domain model and effort required to complete a domain model. Thus two tailed hypothesis i.e. alternate and null hypothesis was formulated. The null hypothesis (H₀) for each dependent variable is: there is no difference between category list technique and noun phrasing technique in terms of completeness and correctness of domain model and required effort. The alternative hypothesis (H₁) is defined as: category list technique produces different quality of domain model, or different effort is required to complete a domain model when compared to noun phrasing technique.

E. Experiment Design

Crossover design is followed in the experiment. Crossover design is a repeated measurement design such that each subject receives different treatments during different time periods. This experiment is conducted in two labs. In first lab, subjects in group A are required to design a domain model for ATM system using noun phrasing technique and group B have to construct a domain model for Internet book store system using category list technique. In second lab, same subjects of group A are required to complete the domain model for Internet book store system using noun phrasing technique and same subjects of group B are required to design a domain model using category list technique for ATM System depicted in table IV. A short presentation was given to the participants to introduce the domain model and its concepts along with the procedure of the

experiment. The hypothesis of the research was not disclosed to avoid any biases later on. The participants were given 40-45 minutes to finish the domain model.

The experiment is performed in supervision of the lab supervisor in both labs. All the required material is provided

to the participants. The participants were required to note the time before starting the experiment and after completion of the experiment. Participants were required to construct the domain model using one technique in first half and alternative technique in second half, respective data is collected.

TABLE I. MEASURES USED TO DERIV DOMAIN MODEL

No#	Measures	Specification
	NCref	Number of correct Concepts in Reference model
	NRref	Number of correct Relationships in Reference model
	NAref	Number of correct Attributes in Reference model
	NGref	Number of correct Generalizations in Reference model
	NCC	Number of correct Concepts in Subjects model
	NCR	Number of correct Relationships in Subjects model
	NCA	Number of correct Attributes in Subjects model
	NCG	Number of correct Generalizations in Subjects model
	NUC	Number of useless Concepts in Subjects model
	NER	Number of extra Relationships in Subjects model
	NEA	Number of extra Attributes in Subjects model

TABLE II. QUALITY MEASURES FOR DOMAIN MODEL

Dependent		Formula
Completeness	Class Completeness $Ccom = NCC / Nc_{ref}$	Completeness= $(Ccom + Rcom + Acom + Gcom) / 4$
	Relationships Completeness $Rcom = NCR / NR_{ref}$	
	Attributes Completeness $Acom = NCA / NA_{ref}$	
	Generalizations Completeness $Gcom = NCG / NG_{ref}$	
Correctness	Number of Useless Concepts NUC	Correctness= $(NUC + NMC + NER + NMR + NEA + NMA + NMG) / 7$
	Number of Missing Concepts $NMC = Nc_{ref} - NCC$	
	Number of Extra Relationship NER	
	Number of Missing Relationship $NMR = NR_{ref} - NCR$	
	Number of Extra Attributes NEA	
	Number of Missing Attributes $NMA = NA_{ref} - NCA$	
	Number of Missing Generalizations NMG	

TABLE III. HYPOTHESIS FOR DOMAIN MODEL CORRECTNESS, COMPLETENESS AND REQUIRE EFFORT

Dependent variable	Null Hypothesis	Alternative Hypothesis
Correct Concepts (CC)	$CC(CLT) = CC(NPT)$	$CC(CLT) \neq CC(NPT)$
Use Concepts (UC)	$UC(CLT) = UC(NPT)$	$UC(CLT) \neq UC(NPT)$
Missing Concepts (MC)	$MC(CLT) = MC(NPT)$	$MC(CLT) \neq MC(NPT)$
Correct Relationships(CR)	$CR(CLT) = CR(NPT)$	$CR(CLT) \neq CR(NPT)$
Extra Relationships(ER)	$ER(CLT) = ER(NPT)$	$ER(CLT) \neq ER(NPT)$
Missing Relationships(MR)	$MR(CLT) = MR(NPT)$	$MR(CLT) \neq MR(NPT)$
Correct Attributes (CA)	$CA(CLT) = CA(NPT)$	$CA(CLT) \neq CA(NPT)$
Extra Attributes (EA)	$EA(CLT) = E(NPT)$	$EA(CLT) \neq E(NPT)$
Missing Attributes(MA)	$MA(CLT) = MA(NPT)$	$MA(CLT) \neq MA(NPT)$
Correct Generalizations(CG)	$CG(CLT) = CG(NPT)$	$CG(CLT) \neq CG(NPT)$
Missing Generalization(MG)	$MG(CLT) = MG(NPT)$	$MG(CLT) \neq MG(NPT)$
Overall Completeness (Com)	$Com(CLT) = Com(NPT)$	$Com(CLT) \neq Com(NPT)$
Overall Correctness (Corr)	$Corr(CLT) = Corr(NPT)$	$Corr(CLT) \neq Corr(NPT)$

Effort	Time(CLT)=Time(NPT)	Time(CLT) ≠Time(NPT)
--------	---------------------	----------------------

1) *Co factors*: There are some extraneous factors that affect the experiment results. These are also known as co founding variables that can also affect the results. In case of influence it becomes difficult to infer that the results are due to the independent variable or due to these co-founding variables. These extraneous factors must be minimized to increase the experiment's effectiveness. In this research students' ability and system complexity are considered as co-founding variables. Subjects of the experiment were choose from the same batch i.e. 4th year students to ensure same level of knowledge and skills regarding domain modeling, however we cannot ignore the fact that students belonging to same class may have different analytical and design skills. These skills would also affect the design of domain model from different complexity systems. Therefore we used a block design of experiment to control the impact of these co-founding variables on the output of the experiment.

Subjects were divided into two blocks according to their grades in software engineering course, so that each group consists of students with almost the same ability as far as software engineering knowledge and skills is concerned.

2) *Learning and fatigue effect*: When subjects deal with the same problem more than once, their response will be better at the second exposure as compared to first one, because human learn from previous experience. As a result any significant changes in the second time can be the effect of practice or learning [14].

In experiment, subjects were required to complete a domain model twice. Different system was used in second half to avoid learning effect.

TABLE IV. EXPERIMENT DESIGN

Lab	Task	Group A	Group B
Lab1	Domain ATM	Category list	Noun phrase
Lab 2	Domain IBS	Noun phrase	Category list

F. Instrumentation

There are three types of instruments associated with experiment: experimental objects, guidelines and measurement [14].

Experimental objects can be a document or source code on which subjects have to work. During experiment planning it is necessary to select appropriate objects i.e. in this experiment; use case description is required for the creation of domain model. In this experiment objects consist of use case description of both software systems (ATM, IBS). Use case description of ATM system [15] and IBS system [16] were selected from literature. A document was provided to students which contains a brief use case description and students were required to design a domain model using pen and papers.

Regarding experiment guidelines, a brief presentation is given to the students in the beginning of the experiment. In which the students were briefly explained about the list of documents provided, the task to be performed, and the submission strategy. A written instructions document is also given which students return at the end of the experiment. The

students were allowed to ask questions before start of the experiment. The students were required to complete the domain model within 45 to 50 minutes. This time selection to construct a domain model is based on the pilot study performed during course work activity.

Measurements contain, documents prepared to collect data and evaluation criteria to compute dependent variables. The use case description documents were prepared and validated. We compared students' domain model with reference model to measure the correctness and completeness of students' domain model. The reference domain model is design by external party, which consists of three researchers having 5 to 10 years of experience in UML and software engineering. The following criteria are followed to evaluate the students' domain model.

- All the concepts were considered correct if different names were used by students for the specific concept in reference model.
- All the relationships belonging to Missing concept in the reference model were considered missing.
- All the relationships of extra concepts were not considered as extra relationships.
- Attribute identified for extra concepts were not considered as extra attributes.
- Attributes were considered as extra identified attributes which are defined in the wrong concept.
- We assume the missing multiplicity to be one.
- In the inheritance, if super class is missing in the students' model, and attributes of super class is correctly defined in the sub class, then those attributes were considered as correctly identified attributes. And missing super class relationships were also considered correct if sub class is correctly associated with the class having direct relationship with super.

G. Analysis Procedure

Data analysis procedure consists of three dependent variables (domain model Correctness, completeness and effort involved to design a domain model), and one independent variable (Method), with two treatments (noun phrase technique and category list technique). The data analysis is performed with help of statistical test. Descriptive statistics presents the initial picture of collected data. Descriptive statistics summarize and presents the quantitative description in an effective way. Some basic descriptive statistics like, mean, standard deviation, minimum and maximum values were presented.

A Mann Whitney U-test was performed for each task related to designing a domain model to compare the means of dependent variables. The dependent variables are not normally distributed therefore we have selected Mann Whitney test which overcame the data normalization assumption.

Three-way ANOVA test is used to analyze combined data collected from lab 1 and lab 2 and extraneous factors which influence the dependent variables. It is used to identify the significance of main effect i.e.: the effect of and interaction between factors [17]. In this experiment two extraneous factors are considered, software systems and students' ability. The purpose of considering these two factors is to analyze the effect of systems' complexity and students' level of understanding on dependent variables and identifying possible interaction between factors.

H. Validity threads

1) Internal validity

Internal validity is concerned with cause-effect relationship among different variables. Internal validity threats can be present when the results of experiment are influenced by extraneous factors like learning and fatigue effect. Learning and fatigue effect is mitigated using cross-over experiment design and two different systems used in different labs.

Although students have same background knowledge but based on their ability subjects were divided into two balance groups according to their grades.

2) Construct validity

Construct validity threats are concerned with the relationship between concepts and construct being studied (correctness and effort). The measurement criteria were briefly explained. We believe that these measurements are reliable. The time factor is directly related to effort being used. Correctness and completeness cover all the domain model elements.

3) External validity

There are two major external validity threats which are related to this experiment, and these threats are usually associated with controlled experiment because of artificial environment used. They are: Are the sample of subjects in this experiment representative of software professionals? Is the material used in experiment representative of real software industry system in terms of complexity and size?

Regarding issue one, 4th year undergraduate student have acceptable knowledge about software engineering and UML modeling. They also practice UML and software engineering concepts during their assignments and projects. Their experience is almost same as junior professionals. Secondly, our purpose is to find the effectiveness of domain modeling techniques which do not need such a high level programming skills and experience. Students do not have exposure about different domain as professionals, but they are familiar about the domain modeling techniques and their usage, which they can apply on any problem domain.

Regarding the second issue, Software systems used in this experiment are small as compared to industrial software systems, because it is not feasible to take large industrial system in limited time [18], but its size and complexity is comparable with other systems used in related experiments [7] [9] [12].

4) Conclusion validity

Conclusions validity threats are related with issues that influence the capability to draw a correct conclusion about experimental hypothesis based on experimental results. Regarding this experiment, appropriate statistical tests were performed to find statistically significant difference. In case where little difference is found but not significant, power analysis was performed to avoid accepting false null hypothesis.

IV. ANALYSIS

Table V and table VI show a Mann-Whitney test results. Overall results show a lack of significant difference between two groups in different dependent variables.

In lab 1, we see a significant difference in the correct concepts (p-value= .021) and correct generalizations (p-value= .039) dependent variables only. From the mean rank of correct concepts show that students produced more correct concepts using category list technique than noun phrase technique.

In lab 2, a significant difference is shown in correct attributes (p-value=.000) and overall completeness (p-value=.009) only. No other dependent variables show significant differences. According to mean rank, subjects produced more correct attributes using noun phrase as compared to category list technique.

Table VI shows the results of overall correctness and effort. As discussed in the dependent variables section that the overall correctness is calculated as average of all the extra dependent variables (concepts, relationships and attributes) and all the missing dependent variables (concepts, relationships and attributes). Lower the overall correctness mean, better will be the quality of domain model.

We can observe from the table VI that those students who used category list technique produced more missing concepts as compare to those who used noun phrase technique in lab 1. However, in lab 2 a significant difference is found in extra attributes (p-value=.000) and missing attributes (p-value=.0000). It can observe from the value of mean rank, that noun phrase technique produced more numbers of correct elements of domain model (concepts, relationships and attributes), However it also produced large number of extra elements in the domain model. No significant difference is found in overall Correctness dependent variable.

We also conduct a power analysis to determine the power of those statistical tests having no significant results. Before accepting null hypothesis we compare minimum effect size required to obtain 80% power with observe effect size. In case of ATM software system, the minimum effect size required to obtain 80% power for overall completeness is 0.512 but the observed effect size is 0.432. Due to the small effect size the observed power is 70%. So we cannot provide any erroneous conclusion about overall completeness of domain model. On the other hand, the observed power of overall correctness is also very small for IBS system. So we cannot reject the null hypothesis.

According to second research question, a significant difference is observed between effort require in term of time. In lab 1 (p-value=.000) and in lab 2 (p-value=.004) were observed for required effort. So from the mean rank we can say that students spent more time in designing a domain model using category list technique as compare to noun phrase technique.

We apply three-way ANOVA test to analyze the combine data of lab 1 and 2 and possible interaction of co-factors, shown in table VIII. In this experiment, System and Ability factors are considered. We observe a significant main effect for the System factor in overall completeness and overall correctness. This significant main effect is in favor of ATM system. The reason of main effect of system may be that students feel more comfortable and performed better in ATM system. We also observe that noun phrase technique produced 6% more complete domain model as compared to category list technique. We do not found any significant interactions between System and Method, Ability

and Method, System and Ability. Which is further elaborated on interaction plot.

Interaction plots highlight interaction in case of nonparallel lines, whereas parallel lines indicate no interaction at all. It can be seen from figure (a) and (b) that subjects with high and low ability performed similarly in both systems in case of completeness of domain model. However it can also be seen that subjects with high ability were able to make a more complete domain model in ATM system using noun phrasing technique. Regarding correctness it is observed from figure (c) and (d) that high and low ability students performed the same whether they used noun phrasing technique or category list in both software system.

Regarding required effort, we observed a significant time difference to complete the domain model in case of both systems. In both software systems students spent more time to complete a domain model using category list technique as compared to noun phrasing technique. This is also observed from interaction plot (e) and (f).

TABLE V. MANN-WHITNEY U TEST OF OVERALL COMPLETENESS

Dependent variable	Technique	Lab 1			Lab 2		
		Mean	Mean Rank	P-value	Mean	Mean Rank	P-value
Correct Concepts	Noun phrase	5.34	28.4	.021	7.08	32.3	.355
	Category List	5.81	38.5		7.41	36.6	
Correct Relationships	Noun phrase	3.42	34.8	.561	5.79	35.5	.664
	Category List	3.12	32.1		5.23	33.4	
Correct Attributes	Noun phrase	3.45	36.1	.251	3.28	45.8	.000
	Category List	2.66	30.8		.882	23.1	
Correct Generalization	Noun phrase	2.36	37.5	.039	1.17	36.1	.445
	Category List	1.72	29.4		.794	32.8	
Overall Completeness	Noun phrase	0.53	36.9	.144	0.29	40.7	.009
	Category List	0.46	30.05		0.23	28.2	

TABLE VI. MANN-WHITNEY U TEST OF OVERALL CORRECTNESS AND EFFORT

Dependent Variable	Technique	Lab 1			Lab 2		
		Mean	Mean Rank	P-value	Mean	Mean Rank	P-value
Useless Concepts	Noun phrase	1.15	36.8	.132	2.50	34.8	.880
	Category List	.88	30.1		2.38	34.1	
Missing Concepts	Noun phrase	2.65	28.4	.021	7.91	36.6	.355
	Category List	2.18	38.5		7.58	32.3	
Extra Relationships	Noun phrase	2.33	36.1	.254	3.02	34.8	.876
	Category List	2.00	30.8		3.05	34.1	
Missing Relationships	Noun phrase	5.57	32.1	.561	13.20	33.4	.664
	Category List	5.87	34.8		13.76	35.5	
Extra Attributes	Noun phrase	4.03	34.9	.404	2.00	45.0	.000
	Category List	3.69	31.0		.352	23.9	
Missing Attributes	Noun phrase	8.54	30.7	.323	12.41	23.1	.000
	Category List	9.33	35.3		15.11	45.8	
Overall Correctness	Noun phrase	4.66	32.97	.822	6.57	30.63	.105
	Category List	3.55	34.03		6.78	38.73	
Effort	Noun phrase	3.60	8.40	.000	23.9	6.36	.004
	Category List	26.8	18.75		41.0	13.8	

TABLE VII. POWER ANALYSIS

Dependent variables	ATM				IBS			
	Observed Power	Mini. Effect size	Effect size	P_ value	Observed Power	Mini. Effect size	Effect size	P_ value
Overall Completeness	0.705	0.512	0.432	.144	--	--	--	.009
Overall correctness	0.891	--	0.077	0.822	0.599	0.526	0.376	.105

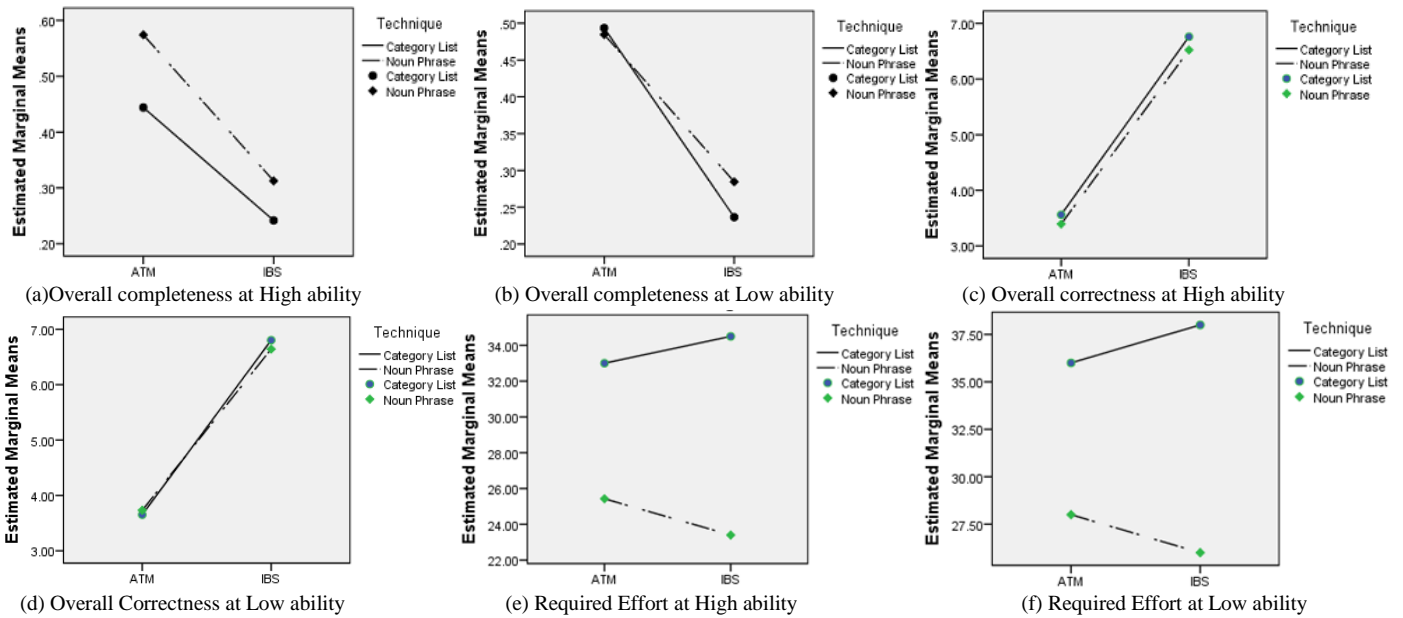


Fig. 1. Analysis results using Interaction Plots

A. Discussion

This experimental study investigates the effectiveness of noun phrase technique and category list technique on the quality of domain model. As already mention that the quality of domain model evaluated in terms of completeness, correctness and effort required for constructing the domain model. We summarize the significant results of a main hypothesis.

RQ1: What is the effect of noun phrase and category list technique on quality of the domain model?

This research question consists of number of hypothesis, and each hypothesis represents different domain model elements.

The statistically significant difference is only found between the number of Correct Concepts (CC) and Missing Concepts (MC) identified by noun phrase technique and category list technique when subjects deal with ATM system. In IBS system, statistically significant difference is found between the number of Correct Attribute (CA), Extra Attribute (EA) and Missing Attributes (MA). Those subjects who used noun phrase technique produce d large number of attributes. Some of the attributes are valid. But most of them are useless. This may be the reason that no specific guidelines were provided to extract attributes from requirement specification using noun phrase technique. After identification of noun and noun phrase, subjects skipped to check each and every noun phrase to decide whether it’s a concept or attribute. In contrast, using category list subjects identified less but valid attributes.

Regarding Overall completeness, a statistically significant difference is found in IBS system only. But both techniques show a lack of domain model completeness. Subjects produced 29% and 23% complete domain model using noun phrase and category list technique respectively in IBS system. On the other hand, subjects produced almost 53%

and 46% complete domain model using noun phrase and category list technique respectively in ATM system. From the combined analysis of both software systems, subjects produced 6% more complete domain model using noun phrase technique as compared to category list.

Regarding overall correctness dependent variables, no statistically significant difference is found in Overall Correctness. On average, subjects produced more extra concepts, relationships and attributes while IBS system using noun phrase technique. So we can say that using noun phrase technique, subjects identified a large number of noun phrases. Some of them were correct and mostly were useless. In addition, satisfactory results were not found while subjects modeled the ATM system. This may be due to the reason that A T M system is common system and easier as compared to IBS system. A little statistically significant difference is found between the overall correctness in IBS system in favor of category list, but due to the low statistical power we cannot reject the null hypothesis about overall correctness.

RQ2: Which domain modeling technique required more effort to design a domain model?

Regarding required effort statistically significant difference is found between both groups to design domain model. Subjects used more time to design domain model using category list technique.

V. CONCLUSION

There are two basic techniques to model problem domain i.e. noun phrase and category list. In category list technique, Larman [1] provided a list of candidate conceptual classes, which consists of many categories that are important to the business information system. Noun phrase technique is a grammatical analysis of use case description to recognize conceptual classes.

To evaluate the impact of category list and noun phrase technique on the quality of domain model an experiment was designed and conducted. The purpose of experiment was to investigate that which technique produces high quality domain model in terms of completeness, correctness and effort required to design a domain model.

According to the statistical tests results, category list technique produced more correct concepts in both software system but the difference is statistically significant only in ATM system. So, we can conclude that category list technique is best for identifying concepts which are important to the business world. It also avoids unnecessary concepts in the problem domain. Noun phrase technique is better for identifying attributes for concepts. Both techniques performed same in case of relationships. Overall subjects produce 6% more complete domain model using noun phrase technique however the results are statistically significant in IBS system only. There is no significant difference found between two techniques regarding overall correctness. Minimal significant difference is found in case of IBS system therefore due to low statistical power we cannot reject the null hypothesis. It is also observed that for known system, it does not matter which technique you are using. We suggest that the combined use of both techniques will lead to high quality domain model.

As a future direction the same experiment need to be executed in an industrial environment for more realistic results. In which professional developers are used as subjects and the scenario is also realistic instead of an exemplary one.

REFERENCES

- [1] C. Larman, Applying UML and Patterns, Prentice-Hall, 2004.
- [2] D. Rosenberg, Use Case Driven Object Modeling with UML – A Practical Approach, APress, Berkeley, USA, 2007.
- [3] D. R. K. Scott, Applying Use Case Driven Object Modeling with UML: An Annotated e- Commerce Example, Addison Wesley , 2001.
- [4] R. Offen, "Domain Understanding Is the Key to Successful System Development," journal of Requirements Engineering , vol. 7, no. 3, pp. 172-175, September 2002.
- [5] J. Evermann and Y. Wand, "Toward Formalizing Domain Modeling Semantics in Language Syntax," IEEE Transactions on Software Engineering, vol. 7, no. 1, p. 21 – 37, January 2005.
- [6] P. Stevens, Using UML: software engineering with objects and components, Pearson Education, 2006.
- [7] T. Yue, L. C. Briand and L. Yvan, "Facilitating the Transition from Use Case Models to Analysis Models: Approach and Experiments," ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 22, no. 1, p. 5, February 2013.
- [8] S. Tiwari and A. Gupta, "Does increasing formalism in the use case template help?," in Proceedings of the 7th India Software Engineering Conference, 2014.
- [9] S. Tiwari and A. Gupta, "A controlled experiment to assess the effectiveness of eight use case templates.," in 20th Asia- Pacific Software Engineering Conference (APSEC), 2013.
- [10] S. Espana, M. Ruiz and A. González, "Systematic derivation of conceptual models from requirements models: a controlled experiment.," in IEEE Sixth International Conference on Research Challenges in Information Science (RCIS), Valencia, 2012.
- [11] B. Anda and D. I. Sjøberg, "Investigating the Role of Use Cases in the Construction of Class Diagrams," journal of Empirical Software Engineering, vol. 10, no. 3, p. 285 – 309, July 2005.
- [12] L. Briand, Y. Labiche and R. Madrazo-R, "An Experimental Evaluation of the Impact of System Sequence Diagrams and System Operation Contracts on the Quality of the Domain Model," in International Symposium on Empirical Software Engineering and Measurement (ESEM), Banff, 2011.
- [13] U. Erra, A. Portnova and G. Scanniello, "Comparing two communication media in use case modeling: results from a controlled experiment.," in ACM- IEEE International Symposium on Empirical Software Engineering and Measurement, 2010.
- [14] C. Wohlin, P. Runeson, M. Host, M. C. Ohlsson, B. Regnell and A. Wesslen, Experimentation in Software Engineering, Springer Science & Business Media, 2012.
- [15] Poul and o. Jeff, Introduction to software testing, 2008.
- [16] Stevens, Perdita and R. J. Pooley, Using UML: software engineering with objects and components., Pearson Education, 2016.
- [17] E. R. Giden, ANOVA Repeated Measures, SAGE Publications.
- [18] G. Scanniello, C. Gravino and G. Tortora, Does the combined use of class and sequence diagrams improve the source code comprehension?: results from a controlled experiment, 12 Proceedings of the Second Edition of the International Workshop on Experiences and Empirical Studies in Software Modeling ACM, 2012.