2004 International Symposium on Empirical Software Engineering

Redondo Beach, California
19–20 August 2004

PROCEEDINGS

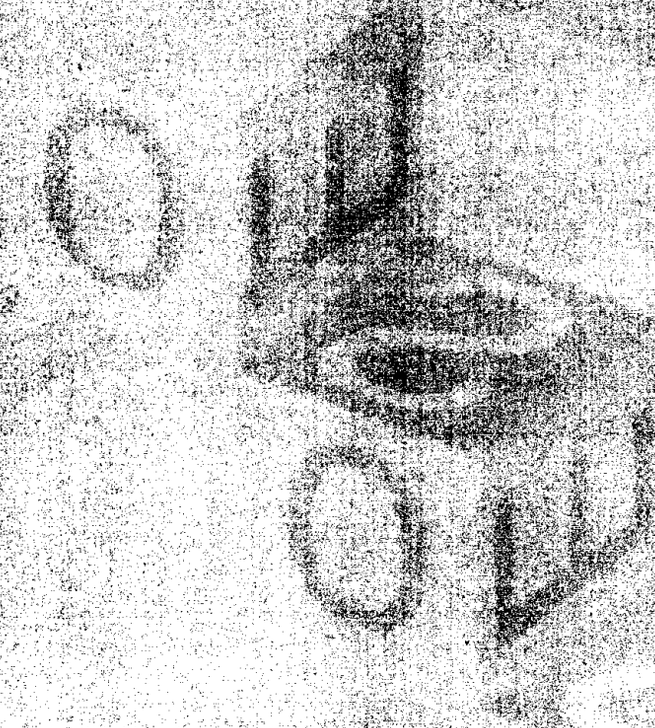
4

ISESE 04

Co-sponsored by
ACM SigSoft
IEEE Computer Society

IEEE COMPUTER SOCIETY

IEEE



Published by the IEEE Computer Society
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314

IEEE Computer Society Order Number P2165
Library of Congress Number 2004104548
ISBN 0-7695-2165-7

ISBN 0-7695-2165-7

IEEE Computer Society Order Number P2165
ISBN 0-7695-2165-7
Library of Congress Number 2004104548

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
http://computer.org/cspress
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

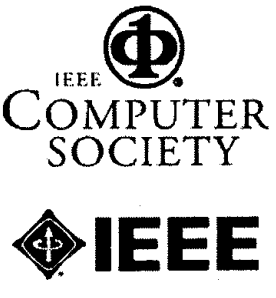
IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Individual paper REPRINTS may be ordered at: reprints@computer.org

Editorial production by Stephanie Kawada

Cover art production by Joe Daigle/Studio Productions

Printed in the United States of America by Applied Digital Imaging



ISESE 2004
Proceedings

2004 International Symposium on Empirical Software Engineering

Table of Contents

ISESE 2004
Proceedings

2004 International Symposium on Empirical Software Engineering

Table of Contents

Session 1B: Software Testing

Session 2A: Programming Practices

Session 2B: Defect Management

Session 3: Infrastructure and Tool Support for Empirical Studies

Session 4A: Cost Estimation

Session 4B: Software Requirements

Session 5A: Predictive Models

Session 5B: Reading Techniques for Inspections

Session 6: Empirical Studies Methodology

Message from the General Chair

Welcome to the third International Symposium on Empirical Software Engineering. This Symposium, created only 2 years ago in Nara, Japan, has become an established part of Empirical Software Engineering International Week (ESEIW). After passing through Rome, Italy last year, the conference will be held for the first time in the United States.

Empirical evaluation of a new technology is a crucial attribute of any science. One must validate a theory with data before accepting the implications of that theory. All too often such validation is missing in our field. Simply claiming a process works is not different from much of the pseudoscience that pervades our world. It is for this reason I believe our Keynote Speaker, Dr. Michael Shermer, although professionally trained as a historian, has much to say of relevance to computer science research.

We have a full two-day program of papers, fast abstracts, poster sessions, and lunch and dinner functions. I would like to thank the Program Co-Chairs, Dr. Forrest Shull and Dr. Natalia Juristo, for putting together an excellent program. I would also like to thank the entire Program Committee and Organization Committee for all their efforts in making this program a success. I also would like to thank both the ACM Special Interest Group on Software Engineering (SIGSOFT) and the Computer Society for their continued support of this activity.

If you are new to the area, I hope you can spend a few days visiting the many sites and attractions in and near Los Angeles. Although southern California does not have the age and history of the venues of the previous two meetings (Nara and Rome), southern California has a very different kind of allure. I hope you find the two days of this Symposium, and any additional meetings you may attend while here, to be fruitful.

Marvin Zelkowitz
University of Maryland
Fraunhofer Center for Experimental Software Engineering at Maryland

Finding "Early" Indicators of UML Class Diagrams Understandability and Modifiability

Marcela Genero and Mario Piatini
ALARCOS Research Group, Department of Computer Science
University of Castilla-La Mancha
Paseo de la Universidad, 4 - 13071 - Ciudad Real (Spain)
{Marcela.Genero, Mario.Piattini}@uclm.es

Esperanza Manso
GIRO Research Group, Department of Computer Science
University of Valladolid
Campus Miguel Delibes, E.T.I.C. - 47011 – Valladolid (Spain)
manso@infor.uva.es

Abstract

Given the relevant role that models obtained in the early stages play in the development of OO systems, in the recent years special attention has been paid to the quality of such models. Adhering to this fact, the main objective of this work is to obtain "early" indicators of UML class diagrams understandability and modifiability. These indicators will allow OO designers to improve the quality of the diagrams they model and hence contribute improving the quality of the OO systems, which are finally delivered. The empirical data were obtained through a controlled experiment and its replication we carried out for obtaining prediction models of the Understandability and Modifiability Time of UML class diagrams based on a set of metrics previously defined for UML class diagrams structural complexity and size. The obtained results, reveal that the metrics that count the number of methods (NM), the number of attributes (NA), the number of generalizations (NGen), the number of dependencies (NDEP), the maximum depth of the generalization hierarchies (MaxDIT) and the maximum height of the aggregation hierarchies (MaxHAgg) could influence the effort needed to maintain UML class diagrams.

Keywords: maintainability, understandability, modifiability, UML, class diagrams, structural complexity, size, metrics, empirical validation, controlled experiments, prediction models.

1 Introduction

In the recent years, software developers have put special attention to guarantee the quality characteristics of OO systems, such as understandability and modifiability, since the initial stages of their life cycle [1,2,3,4]. Recently, paradigms such as Model-Driven Development [5] and the Model-Driven Architecture [6] have emphasized the importance of "good" models from the beginning of the life cycle. For that reason, the main focus must be on the quality of models obtained in these "early" stages. As class diagrams are the backbone of OO systems, we decided to investigate the quality characteristics, such as understandability and modifiability of UML class diagrams[1]. So we began investigating those works related to metrics that can evaluate quality characteristics of UML class diagrams in an objective way. After a deep research of the existent metrics we defined and validated a set of metrics for measuring UML structural complexity and size, due to the use of UML relationships [7,8]. Table 1 shows a brief description of those metrics.

As it is well known in the field of software measurement, if we want metrics that measure internal attributes (size, structural complexity, etc.) to be useful, it is necessary that they can be used to predict some external attribute of quality, such as understandability and modifiability. Therefore, our main motivation since the last three years has been to investigate, through experimentation, whether the metrics we proposed could be good predictors of UML class diagram understandability and modifiability. If this is corroborated by several empirical studies, we will really have obtained early indicators of class diagram maintainability. These indicators will allow OO software designers to make better decisions early in the OO software development life cycle, thus contributing to the development of better quality OO software.

This paper has two main objectives:

- To find a prediction models which relates the metrics shown in table 1 with maintainability measures, such as Understandability and Modifiability Time. This was done using the data obtained through a controlled experiment carried out with students at the University of Seville in Spain.
- To evaluate the predictive accuracy of the obtained models using the data obtained in a replication of the experiment undertaken with other similar group of students of the same university.

This paper starts with a description of related work. Following that, in section 3 a description of a controlled experiment we carried out is presented. A replica of this experiment is presented in section 4. Section 5 provides the data analysis and interpretation, and finally the last

[1] We selected understandability and modifiability because according to the ISO 9126 [9] are the major determinants of maintainability. Moreover, this fact has been demonstrated by several empirical studies in the software engineering field.

0-7695-2165-7/04 $20.00 © 2004 IEEE

section presents some concluding remarks and outlines directions for future research activities.

Table 1. Metrics for UML class diagram size and structural complexity

Type of Metrics	Metric definition
Size metrics	Number of Classes (NC). The total number of classes.
	Number of Attributes (NA). The total number of attributes.
	Number of Methods (NM). The total number of methods.
Structural complexity metrics	Number of Associations (NAssoc).The total number of associations.
	Number of Aggregations (NAgg). The total number of aggregation relationships (each "whole-part" pair in an aggregation relationship).
	Number of Dependencies (NDep). The total number of dependency relationships.
	Number of Generalisations (NGen). The total number of generalisation relationships (each "parent-child" pair in a generalisation relationship).
	Number of Generalisation hierarchies (NGenH). The total number of generalisation hierarchies.
	Number on Generalisation hierarchies (NAggH). The total number of aggregation hierarchies (whole-part structures.)
	Maximum DIT (MaxDIT). It is the maximum DIT value obtained for each class of the class diagram. The DIT value for a class within a generalisation hierarchy is the longest path from the class to the root of the hierarchy.
	Maximum HAgg (MaxHAgg). It is the maximum HAgg value obtained for each class of the class diagram. The HAgg value for a class within an aggregation hierarchy is the longest path from the class to the leaves.

2. Related work

As some studies which have reviewed the state of the art of empirical studies related to OO measures reveal [3,11,12,13] the dependent variables investigated are either fault proneness (probability of fault detection), the number of faults or changes in a class, the effort of various development activities, or expert opinion about psychological complexity of a class.

Related to maintainability characteristics as dependent variables (the subject we are occupied with in the present study) several works exist [1,13,14,15], which have proposed prediction models for maintenance tasks. But in most of these studies the measurement of the independent variables was performed from the source code and not from UML class diagrams, and for this reason the predictions have been made later in the development. Despite this fact, Briand and Wüst [2,3] and Card et al. [16], among others, highlighted that the earlier the measurement is taken the better.

This work is part of a project we have been developing during the last three years with the aim of looking for early indicators of UML class diagram maintainability. Here, we will briefly summarise our previous work:

- First, we thoroughly analysed the existent measures that could be applied to class diagrams at a high-level design stage, and proposed new ones which measure the structural complexity and size of class diagrams, due to the usage of UML relationships (see table 1) [7,8].
- In [17] we presented an experiment in which the subjects were given 24 class diagrams and they had to subjectively evaluate some maintainability sub-characteristics. Even though the opinion of the subjects is in nature subjective, the preliminary findings were encouraging. All of the metrics we proposed seemed to be related to maintainability sub-characteristics.
- In [18] we presented an experiment and its replica where the subjects were given nine class diagrams and they had to modify them to achieve some new requirements. In this case, we found that metrics related to aggregation and generalisation relationships are highly correlated to modifiability correctness and completeness.
- In [19] we described an experiment where the subjects were given nine class diagrams and had to modify them according to three new requirements, and had to write down the time used in those modifications. As a result, we found that the maintenance time seemed to be correlated with all the metrics except those related to the number of dependencies.

Moreover, the current study improves the previous one. As in [20] we have searched for Principal Components (PCs) which summarize the information of the set with eleven class diagram metrics. In addition we have exploited these results studying the Understandability Time, Modifiability Time and total time prediction models using three groups of independent variables. The first one with the eleven structural complexity metrics, shows collinearity problems. The second one with only the PCs, which solves the collinearity problems but can make understandability difficult. And the last group with only the most relevant structural complexity metrics in PCs, because the initial dimensionality (eleven metrics) is reduced without losing relevant information and we can understand them easier than

PCs, furthermore some collinearity problems could be avoided.

The predictive accuracy and reliability of the models studied in [21] were over the same data set used to adjust the model, (section 5.4.3), that is usual. But in this work it is considered another data set, that was obtained through the replication) for this purpose, which is the best approach to estimate the prediction model reliability.

3. Experiment description

We decided to carry out this experiment trying to improve some issues not covered in the previous one:

- Increasing the number of subjects.
- Increasing the quantity of class diagrams.
- Considering "real" UML class diagrams, taken from real projects, exams, etc.
- Considering class diagrams of different size and complexity, covering a wide range of metrics values.
- Supervising the execution of the experiment.
- Improving the experimental tasks.

For the sake of brevity in this section we will only outline the main aspects of the experimental process [22]:

- The main objective of the experiment is to investigate whether the metrics we proposed [7,8] for the structural complexity and size of UML class diagrams can be used as early indicators of their understandability and modifiability.
- The subjects were seventy-two students enrolled in the fourth-year of Computer Science at the Department of Computer Science at the University of Seville in Spain. They were chosen for convenience, i.e., the subjects were undergraduate students which were taking a second Software Engineering course, when the experiment was run. They have approximately one year and a half of experience in designing UML class diagrams, but they have two years of experience in the OO paradigm, because they begin programming with JAVA since the first year of the degree.
- The independent variables were the structural complexity and size of UML class diagram. They were measured through the metrics we proposed (see table 1). The dependent variables are two maintainability sub-characteristics: understandability and modifiability. These dependent variables were measured through the time the subjects spent doing the required tasks ("Understandability Time" and "Modifiabiality Time" respectively).
- The experimental objects were 24 UML class diagrams[2]. Since we wanted to have objects of different complexity we designed them covering a wide range of the metrics values. But really, it is impossible to cover all of the possible metrics value. For that reason once we have chosen the diagrams we carried out a hierarchical clustering to group them into three groups: "Low complexity" (L), "Medium complexity" (M) and "High complexity" (H). This clustering was run using the metrics values, and helped us to know if we have to reduce or increment the metrics values trying to have 8 diagrams of each group (see table 2). In that way we obtained an objective classification of the diagrams, which we called "DiagType".
- We formulated the main hypotheses we wish to test:
  - **Hypotheses 1:** $H_{0,1}$: The structural complexity and size metrics are good predictors of Understandability and Modifiability Time. $H_{1,1}$: $\neg H_{0,3}$
- We also want to investigate the coherence between the objective and subjective classification of the class diagrams. Furthermore it is interesting to study if there are some maintainability aspects, such as Understandability or Modifiability Time, strongly correlated with the subjective classification. Therefore, we formulate the following hypotheses:
  - **Hypotheses 2:** $H_{0,2}$: the subjective classification of the class diagrams (SubComp) is not correlated with the Understandability and Modifiability Time // $H_{1,2}$: $\neg H_{0,2}$
  - **Hypotheses 3:** $H_{0,3}$: the subjective classification of the class diagrams (SubComp) is not correlated with objective classification (DiagType). // $H_{1,3}$: $\neg H_{0,3}$
- Moreover, we wish to ascertain if there exists correlation between the Understandability and the Modifiability Time of the class diagrams. Because if there exists correlation, the first one could be a good predictor of the second one. Therefore, we formulated the following hypotheses:
  - **Hypotheses 4: $H_{0,4}$:** the Understandability Time is not correlated with the Modifiability Time. // $H_{1,4}$: $\neg H_{0,4}$
- Each diagram had an enclosed test that included a brief description of what the diagram represented, and three tasks:
  1. Understandability tasks: where the subjects had to answer a questionnaire (4 questions) that reflected whether or not they had understood each diagram, and they also had to write down how long it took to answer the questions. The

[2] All the experimental material can be found on http://alarcos.inf-cr.uclm.es.

"Understandability Time", expressed in seconds, was obtained from that.

2. Modifiability tasks: Where the subjects had to modify the class diagrams according to four new requirements, and specify the start and end time. The difference between the two times is what we call "Modifiability Time" (expressed in seconds), which includes both the time spent analysing what modifications had to be done and the time needed to perform them. The activities that the subjects have to carry out are considered as "enhancive maintenance", according to the types of software maintenance proposed by Chapin et al. [23]. The tasks for each class diagram were similar, including adding or replacing attributes, methods, classes, etc.
3. Each subject had to rate the complexity of each diagram using a scale consisting of five linguistic labels (see table 3). This measure was called "SubComp".

Table 2. Metric values for each UML class diagram

DIAGRAMS	METRICS										
	NC	NA	NM	NAssoc	NAgg	NDep	NGen	NGenH	NAggH	MaxDIT	MaxHAgg
L1	6	28	52	5	0	0	0	0	0	0	0
L2	7	23	60	3	1	1	2	1	1	1	1
L3	7	19	39	3	3	1	1	1	1	1	3
L4	6	18	38	3	2	0	0	0	2	0	1
L5	6	11	39	2	3	0	1	1	3	1	1
L6	5	9	18	4	3	0	0	0	1	0	2
L7	8	17	24	2	3	2	1	1	1	1	2
L8	7	25	85	4	2	0	1	1	1	1	2
M1	9	31	78	6	3	1	0	0	2	0	2
M2	14	27	60	11	0	0	6	1	0	2	0
M3	17	33	66	12	5	0	4	1	3	2	1
M4	10	33	83	6	5	1	1	1	3	1	2
M5	9	42	89	10	1	0	2	1	1	1	1
M6	13	34	57	3	4	1	6	3	2	1	3
M7	13	39	96	6	5	1	4	2	2	1	4
M8	11	46	79	8	2	1	2	1	2	1	1
H1	30	54	128	12	7	3	17	1	3	4	4
H2	52	76	35	15	19	8	21	7	2	4	7
H3	39	65	71	11	6	8	23	2	3	3	2
H4	24	57	232	16	8	1	5	2	5	1	2
H5	19	56	147	13	6	3	3	2	4	1	2
H6	28	49	111	10	4	2	14	3	1	3	2
H7	28	39	111	12	4	2	10	1	1	2	1
H8	27	33	82	7	5	2	15	3	1	2	1

Table 3. Linguistic labels for complexity[3]

Very simple	Slightly simple	Normal	Slightly complex	Very complex

[3] For carrying out the data analysis we assigned numbers to each linguistic level in the following way: "Very simple" correspond to 1 and "Very complex" correspond to 5.

- The subjects were given an intensive training session before the experiment took place. In this session we gave them one test similar to those we used in the experiment. We explained to them the tasks they had to carry out. Once this test used as example was finished we collected the data and checked if they really have understood the tasks. After the training session the experiment was executed. The subjects were joined in a room and supervised by a professor. We chose a simple between-subjects[4] and balanced design experiment, i.e., each subject had to solve only one test (assigned in a randomly way) and each test was handed out to the same number of subjects. We are conscious that a between-subjects design may lead to more variability due to great variation in subject's skill level. For the replication, we will measure the skill level through a training session. Nevertheless, in the current experiment we through a debriefing questionnaire related to the subjects'personal experience we detected that their skill level were similar.
- The professor who monitored the experiment explained to them how to carry out the tests. They had a total of two hours to solve the tests. We called the data obtained in this experiment "SE-1" data.

4. Replication of the experiment

The experiment was replicated with other group of twenty eight students. They were also undergraduate students enrolled in the fourth-year of Computer Science at the University of Seville. Therefore, the characteristics of the subjects were similar. The data obtained in this replication, was called "SE-2" data.

5. Data analysis and interpretation

The SE-1 data obtained in the experiment previously described and the SE-2 data obtained through its replication, are the data we will analyse.
For analysing the empirical data (SE-1 data) we will carry out the following steps:

1. First we will do a descriptive and exploratory study (section 5.1).
2. Then we will study what time, Understandability or Modifiability, has the higher correlation with the class diagram subjective complexity (SubComp), hypotheses 2. That is, which effort time has influenced on the students to classify them. Furthermore, we want to know how much the objective classification is according to the subjective one, hypotheses 3 (section 5.2).
3. Afterwards we will study the correlation between the Understandability Time and the Modifiability Time, in order to search how much the first one could to have an influence on the second one, hypotheses 4 (section 5.3).
4. We will model the relationship between the explanatory variables and each dependent variable using a multivariate linear model (section 5.4), obtaining prediction models for the Understandability and Modifiability Time. Firstly we will consider the SE-1 data to adjust the proposed multivariate linear models. As part of this last study we will test the validity of the models assumptions, to see if the residuals obeyed the hypotheses of the model normality, independence, etc. (section 5.4.1).
5. Moreover, we will assess the predictive accuracy and reliability of the models (section 5.4.2), that is, Will the chosen models give accuracy predictions for a new class diagram?. Generally the answer to this question is analysed over the same data used to adjust the model, but in this work it is considered another data set (SE-2) for this purpose, which is a good approach to answer the question.

5.1. Descriptive and exploratory studies

At the beginning, the group SE-1 has 72 subjects, but we have excluded the subjects SE-1-12 and SE-1-28 because their results have errors in the dependent variable measures. The descriptive statistics of the dependent variables, excluding these two cases, are presented on table 4. Looking at the interval (Mean ± 3*SE) we have found one understandability outlier and one modifiability outlier, SE-1-11 and SE-1-4 respectively. When these two outliers were excluded the skewness and the kurtosis improved, so the variable distributions were closer to the normal distribution. This condition of normality will be relevant when selecting the regression models for predicting the dependent variables.

We have observed that the descriptive statistics of Understandability Time are less than the Modifiability Time ones. This fact can reveal a different behaviour of these variables, in fact the mean time to do one Understandability task (135.070 /4 = 33.768 seconds) was approximately a third of the mean time to do a Modifiability task (381/4=95.350 seconds).

[4] We selected a between-subject design, because the statistical tests we decided to apply in the analysis require data independence. Due to space constraints, we cannot explain how we have distributed the data for carrying out the analysis. But we could explain this by request.

Table 4. Descriptive statistics of dependent variables in SE-1 data

	SE-1 (n=70)					
Dependent variables	**Mean**	**SE**	**Median**	**IQR**	**Skewness**	**Kurtosis**
Understandability Time	139.670	67.594	130.00	101.500	0.931	0.959
Modifiability Time	397.260	206.878	361.000	272.500	1.784	5.635
	SE-1 (n= 68)					
	Mean	**SE**	**Median**	**IQR**	**Skewness**	**Kurtosis**
Understandability Time	135.070	61.307	122.500	96.250	0.597	-0.442
Modifiability Time	381.280	173.210	359.500	227.250	0.853	0.617
SE: Standard Error IQR: Inter-Quartile Range						

Table 5. Correlation between subjective and objective classifications (*) significant at level 0.05

Variables	$\tau_{kendall}$	p-value	size
SubComp and Understandability Time (Hypotheses 2)	0.242	0.025 (*)	62
SubComp and Modifiability Time (Hypotheses 2)	0.147	0.145	62
SubComp and DiagType (Hypotheses 3)	0.539	0.000(*)	64

5.3. Testing hypotheses 4

This exploratory analysis tries to discover if there exists correlation between the time that is necessary to understand a class diagram and the time to modify it, which is the hypotheses 4.

Using the Pearson coefficient the result was not significant at level 0.05 (r= 0.069 p= 0.576 n=68). The test power to detect an effect size of 0.30 was 0.71 ,which means that the error to accept no linear correlation between Understandability and Modifiability when there exists is 0.29.

When the correlation was examined by DiagType variable, the results were non-significant (p-values greater than 0.20), so the DiagType seems to be a factor which has not influenced in the correlation between Understandability and Modifiability Time.

In conclusion, it seems that there is no correlation between Understandability and Modifiability Time and the error to accept this fact ($H_{0,4}$) is 0.29.

5.4. Testing hypotheses 1

We have selected the following Multivariate Linear model to test the main goal of this study (hypotheses 1):

$$Y = \sum_{j=1}^{r} \beta_j X_j + \varepsilon$$

Where Y is one of the dependent variables, and X_j are the independent or explanatory variables that explain Y significantly, and ε are the residuals with $N(0,\sigma)$ distribution. Note, that we consider the model without intercept, due to they obtain better results. Moreover as the considered metrics are highly non-normal it would make sense to transform time and size via logarithmic before fitting the regression model.

In software engineering experimentation [3], multivariate analysis is commonly used, because it looks at the relationships between independent and explanatory variables. But it also considers the former by way of combination as covariates in a multivariate model, in order to explain in a better way the variance of the dependent variables, and ultimately obtain accurate predictions.

The criterion used to select the model was the simpler and the easier to interpret, with less p-value of F-test and the best goodness of fit (R^2). When the models were selected following these rules, it was necessary to look for the outliers, influential points and collinearity.

Through a Principal Component Analysis we have obtained that only seven of the defined metrics are relevant: NC, NM, NGen, NAgg, NAggH, MaxDIT, MaxHAgg. In order to exploit these results we have studied the Understandability Time and Modifiability Time prediction models using two groups of independent variables:

- Group A. The eleven structural complexity metrics as explanatory variables, which is the main aim of this study.
- Group B. The seven metrics more relevant in the PCA, because the dimensionality is reduced and they have relevant information, and some collinearity problems of the group A could be avoided.

The models selected are shown in tables 7 and 8, which show the ANOVA results to contrast the hypotheses: $H_{0,1}$: $\beta_j = 0 \ \forall j$ // $H_{1,1}$: $\neg H_{0,1}$

The information that shows these tables refers to:

- Adjusted linear models grouped into A and B (GA and GB, respectively).
- p-value, if it is lower than 0.05 the model is accepted.
- R^2 goodness of fit, which refers to the rate of the dependent variable variability explained by the model.

Looking at table 6, we can conclude that The independent variables of Understandability models were NA, NDep and MAxHagg in GA group and NM and

MAxHagg in GB group. The MaxHagg metric was the common independent variable between the two groups. The model significances were p=0.000. The GB model had the minimum R^2 (77.6%) and the model derived from it (GB-Lg), using logarithmic transformations, had the maximum R^2 (98.1%).

From table 7 we can conclude that the independent variables of Modifiability models were NA, NGen and MAxDIT in GA group, and NM with MAxHagg in GB group. The model significances were p=0.000, the minimum R^2 was 68.8% 8 (GB model) and the maximum 84.7% (GA-Lg model).

The GB models used the same metrics as independent variables (NM and MaxHagg), that means, the number of methods (class diagram functionality) and the maximum height of aggregation hierarchies can explain Understandability and Modifiability Time. The GA models had the number of attributes (NA) as common variable.

To detect collinearity we can look at the condition number (CN), that is the degree of ill conditioning of the explanatory variables; it is calculated using the eigenvalues of the explanatory variable matrix. The values of CN[5] of the selected models pointed out weak dependencies, which produced the negative coefficients in group A models. This fact can difficult the model understanding, so we have tried to develop GA models without negative coefficients (combining or adjusting independent variables) but the results were worst than the selected ones. Finally, when we used the logarithmic transformation the selected models improved the results, not only the negative coefficients have another meaning, furthermore others model properties as error normality, improved.

The SPSS [24] provides some statistics that can detect the influential points, such as Cook's[6] distance, adjusted difference (DFFIT)[7], etc. [2]. The outliers can be detected looking at the residual values. Using these statistics we detected two outliers in Modifiability Time model GA-Lg (SE-1-10 and SE-1-2) which were excluded. The considered models did not present influential points.

5.4.1. Model Validation. One of the threats to conclusion validity of the empirical studies is violating assumptions of statistical tests. That is why we have studied linear model validation about normality, independence and homogeneity of variance (Kleinbaum et al., 1987).

- The hypotheses to contrast the normality of residuals (e_i) are:

  H_0: $e_i \rightarrow N(\mu,\sigma)$ // H_1: $\neg H_0$

  These hypotheses have been tested using the Kolmogorov-Smirnov test. Looking at the p-values of the table 8 we can see one Understandability Time model (GB model) that could have normality problems[8], resolved with logarithmic transformation.
- The Durbin-Watson test contrasts the independence of residuals (e_i) [26]:

  H_0: (e_i) do not have first-order autocorrelation // H_1: $\neg H_0$

  The results were non-significant with respect to Understandability Time models, at level 0.05. The Modifiability models GA and GB were not conclusive, but the transformed model GA-Lg was non-significant.
- To explore the homoscedasticity (homogeneity of variance), we looked at the scatter diagrams of standardized residuals against standardized predicted values, and they did not show any deviation form.

Summarizing, the GB-Lg Understandability Time model and GA-Lg Modifiability Time model have the best goodness of fit and does not present any problems about linear model assumptions.

5.4.2. Model predictive accuracy. In order to evaluate a model it is necessary to consider not only the goodness of fit, the p-value and the simplicity, there is another important quality aspect of the models with respect to their prediction accuracy. The MMRE and PREd(l) are the more used measure of prediction accuracy [27]. Their definitions are:

$$MRE_i = \left| \frac{Observed\_Time - Predicted\_Time}{Observed\_Time} \right|$$

$$MMRE = \frac{1}{n} \sum_{i=n}^{i=1} = MRE_i$$

Where i represents each point of the data set. The prediction level is defined as Pred(l) = k/n %, where k is the number of the points with MREi lower than l; it means that the prediction at l level measures the estimation percentage with error less than l.

Generally, these estimators are evaluated over the same data used to adjust the model, but this study has considered another data set (SE-2) for this purpose, which descriptive statistics are in table 10.

Looking at tables 4 and 9 we noticed that the range and the dispersion of the dependent variable values in the SE-2 data is a bit greater than in the SE-1 data.

As tables 10 and 11 show the best models, from the point of view of accuracy prediction are Understandability Time GA model, with MMRE 0.340

[5] Weak dependencies are associated with the condition number around 5 or 10, moderate to strong relations are associated with the condition number of 30 to 100.

[6] The change in the regression coefficients when a point is excluded. A general role is to check the model if a point has taken a value greater than 1.

[7] The standardized change in a point prediction when this point is excluded. The role is to consider an influential point if the DFFIT is greater than 2/sqrt(p/n), if p is the number of independent variables and n is the sample size.

[8] The usual significance level, in this kind of hypothesis, is about 0.15 because it permits to improve the test power.

and Pred(0.30) equal to 52.8, and Modifiability Time GA model with MMRE 0.427 and Pred(0.30) equal to 62.2.

Table 6. Understandability Time (UT) Models

	Adjusted linear model (forward)	p-value	R^2
Group A	UT= 3.444*NA – 18.552*NDep + 14.991*MaxHAgg	0.000	0.792
Group B	UT= 0.942*NM + 23.094*MaxHAgg	0.000	0.776
Group B-Lg	Lg(UT)= 1.075lg(NM) + 0.384*Lg(MaxHAgg)	0.000	0.981

Table 7. Modifiability Time (MT) Models

	Adjusted linear model (forward)	p-value	R^2
Group A	MT= 8.918*NA-30.245*NGen + 139.850*MaxDIT	0.000	0.775
Group A-Lg	MT= 251.325*Lg(NA)	0.000	0.847
Group B	MT=2.319*NM+ 68.743*MaxHAgg	0.000	0.688

Table 8. Residual Normality test (Kolmogorov-Smirnov test) (*) significant at level 0.05

p-values			
Understandability Time		**Modifiability Time**	
Group A	0.200(*)	**Group A**	0.200(*)
Group B	0.065	**Group A-Lg**	0.200(*)
Group B-Lg	0.20(*)	**Group B**	0.200(*)

Table 9. Descriptive statistics of dependent variables in SE-2 data (n=28)

	Mean	SE	P_{25}	Median	P_{75}
Understandability Time	155.429	88.876	98.000	132.500	181.500
Modifiability Time	394.250	353.504	255.000	302.500	398.500

Table 10. Prediction accuracy of the selected Understandability Time models in SE-2 data

Model	MMRE	P_{25}	P_{50}	P_{75}	PRED (0.30)
Group A	0.340	0.135	0.274	0.527	**52.8%**
Group B	0.369	0.101	0.355	0.615	51.1%
Group B-Lg	0.458	0.085	0.313	0.708	51.1%

Table 11. Prediction accuracy of the selected Modifiability Time models in SE-2 data

Model	MMRE	P_{25}	P_{50}	P_{75}	PRED (0.30)
Group A	0.427	0.169	0.261	0.664	**62.2%**
Group A-Lg	0.453	0.126	0.283	0.571	52.0%
Group B	0.462	0.219	0.427	0.624	35.7%

5.5 Conclusions of the data analysis

Summarising, the findings obtained through the current study are the following:

- The models with forward selection and without intercept were the best.
- The GB-Lg model (obtained using logarithmic transformations) was the best for Understandability Time, that is a ratio between the metrics NM and MaxHagg of UML class diagrams can adjust and predict Understandability Time with good accuracy (see tables 6 and 10).
- The GA-Lg model was the best for Modifiability Time. It means that a ratio between the metrics NA and NGen (size and abstraction complexity of class diagrams) can adjust and predict Modifiability Time with good accuracy (see tables 7 and 11).
- The two types of class diagram classifications, the subjective and the objective one are correlated, furthermore the understandability effort has influenced in the student subjective classification.
- The Understandability Time is not correlated with the Modifiability Time.

6. Conclusions

Pursuing the objective of obtaining good predictors of the Understandability and Modifiability Time of UML class diagrams, we have presented in this work:

1. A controlled experiment and its replication, carried out with students enrolled in the fourth-year of Computer Science at University of Seville (see section 3 and 4).
2. A thoroughly data analysis, through which we obtained prediction models based on some of the metrics we proposed for the size and structural complexity of UML class diagrams (see section 5).

After a multivariate regression analysis, followed by validation of the obtained models, we can conclude that the obtained multivariate linear models have proved that the Understandability and Modifiability Time are related -to some extent- to the size and structural complexity measures we previously defined (see table 1).

The obtained results reveal that the metrics related to the number of classes (NC), the number of associations (NAssoc), the number of aggregations (NAgg), the number of aggregations hierarchies (NAggH) and the number of generalizations hierarchies (NGenH) do not seem to have influence on maintenance effort.

These findings could be very valuable when predicting the maintenance effort of OO software products, one of the biggest concerns in software organisations.

From the results presented in this study, we may conclude that there is reasonable chance that useful class diagram Understandability and Modifiability Time models could be built at the initial phases of OO systems life cycle, e.g. when choosing between two semantically equivalent design alternatives, thus allowing OO software designers to take better decisions early in the OO software development life cycle. Nevertheless we do not believe that universally valid quality measures and models can be devised at this stage. Therefore, the focus of our multivariate analysis is to obtain an initial assessment of the feasibility of building class diagram maintainability prediction models based on early metrics. However, as Briand and Wüst (2001) remarked, early analysis and design artefacts are, by definition, not complete, and only represent early models of the present system to be developed. For this reason, the use of predictive models based on early artefacts and their capability to predict the quality of the final system still remain to be investigated.

This experiment has resulted in some interesting and surprising findings about the determinants of maintenance effort in UML class diagrams. However we have identified a number of weaknesses in the study, particularly with respect to external validity. The results of this experiment should therefore be interpreted as preliminary findings only, which need replication and corroboration. We propose to conduct further research in two directions:

- Experimental research: It is our belief that it is necessary to make a family of experiments to increase the external validity of the results to the extent that the conclusions currently presented can be generalized. Such a family of experiments should also use professionals as subjects and different experimental models and different experimental tasks (class diagrams and maintenance changes taken from practice). Besides we are conscious of the necessity to make laboratory packages with the information of the empirical studies, to encourage their external and independent replication and obtain a body of knowledge about the utility of metrics. This will eventually contribute to build a set of UML class diagram metrics and quality prediction models that allow software designers to make better decisions in the early phases of software development. After all, this is the most important goal for any measurement proposal to pursue if it aims to be useful.
- Field studies: we also plan to conduct a number of field studies on the effect of size and structural complexity on maintenance of UML class diagrams in practice. This will involve observations of maintenance practices in OO development environments. Data collected will include:
  - Size and structural complexity of models, using the full range of metrics defined
  - Frequency and type of maintenance changes required: this will enable evaluation of the stability of models.
  - Effort required to implement changes: this will enable evaluation of analysability and modifiability of models.

Acknowledgements

This research work is part of the MESSENGER project (PCC-03-003-1), financed by the "Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla - La Mancha (Spain)" and of the CALIPO project (TIC2003-07804-C05-03), financed by the "Dirección General de Investigación of the Ministerio de Ciencia y Tecnología (Spain)".

We want to thanks Isabel Ramos from the University of Seville for allowing us to perform the experiment with their students. Moreover, we want thanks the anonymous reviewers for their fruitful comments that allow us to improve the present work.

References

1. L. Briand, C. Bunse, and J. Daly, "Controlled Experiment for evaluating Quality Guidelines on the Maintainability of Object-Oriented Designs", *IEEE Transactions on Software Engineering*, 27(6), 2001, pp. 513-530.
2. L. Briand, and J. Wüst, "Modeling Development Effort in Object-Oriented Systems Using Design Properties", *IEEE Transactions on Software Engineering*, 27(11), 2001, pp. 963-986.

3. L. Briand, and J. Wüst, "Empirical Studies of Quality Models in Object-Oriented Systems", *Advances in Computers*, Academic Press, Zelkowitz (ed.), 59, 2002, pp. 97-166.
4. J. Bansiya, and C. Davis, "A Hierarchical Model for Object-Oriented Design Quality Assessment", *IEEE Transactions on Software Engineering*, 25(4), 2002, pp. 4-17.
5. C. Atkinson, and T. Kühne, "Model-Driven Development: A Metamodeling Foundation", *IEEE Software*, 20(5), 2003, pp. 36- 41.
6. Object Management Group. MDA-The OMG Model Driven Architecture. Available: http://www.omg.org./mda/, August 1st, 2002.
7. M. Genero, "Defining and Validating Metrics for Conceptual Models", *Ph.D. Thesis*, University of Castilla-La Mancha, 2002.
8. M. Genero, M. Piattini, and C. Calero, "Early Measures For UML class diagrams", *L'Objet*, 6(4), Hermes Science Publications, 2000, pp. 489-515.
9. ISO/IEC 9126-1.2., "Information technology-Software product quality – Part 1: Quality model", 2001.
10. V. Basili,, F. Shull, and F. Lanubile, "Building Knowledge through Families of Experiments", *IEEE Transactions on Software Engineering*, 25(4), 1999, pp. 435-437.
11. K El-Emam, "Object-Oriented Metrics: A Review on Theory and Practice", *NRC/ERB 1085*, National Research Council Canada, 2001.
12. I. Deligiannis, M. Shepperd, S. Webster, and M. Roumeliotis, "A Review of Experimental into Investigations into Object-Oriented Technology", *Empirical Software Engineering*, 7(3), 2002, pp. 193-231.
13. W. Li, and W. Henry, "Object-Oriented Metrics that Predict Maintainability", *Journal of Systems and Software*, 23(2), 1993, pp. 111-122.
14. R. Harrison, S. Counsell, and R. Nithi, "Experimental Assessment of the Effect of Inheritance on the Maintainability of Object-Oriented Systems", *Journal of Systems and Software*, 52, 2000, pp. 173-179.
15. F. Fioravanti, and P. Nesi, "Estimation and Prediction Metrics for Adaptive Maintenance Effort of Object-Oriented Systems", *IEEE Transactions on Software Engineering*, 27(12), 2001, pp. 1062-1083.
16. D. Card, K. El-Emam, and B. Scalzo, "Measurement of Object-Oriented Software Development Projects", *Software Productivity Consortium NFP*, 2001.
17. M. Genero, J. Olivas, M. Piattini, and F. Romero, "Using metrics to predict OO information systems maintainability", CAISE 2001, *Lecture Notes in Computer Science*, 2068, Interlaken, Switzerland, 2001, pp. 388-401.
18. M. Genero, L. Jiménez, and M. Piattini, "A Controlled Experiment for Validating Class Diagram Structural Complexity Metrics", 8th International Conference on Object-Oriented Information Systems (OOIS'2002), *Lecture Notes in Computer Science*, 2425. Bellahsene, Z., Patel, D., Rolland, C., (Eds.). Springer-Verlag, 2002, pp. 372-383.
19. M. Genero, MªE. Manso, M. Piattini, and G. Cantone , "Building UML Class Diagram Maintainability Prediction Models Based on Early Metrics", *9th International Symposium on Software Metrics* (Metrics 2003), Proceedings IEEE Computer Society, 2003, pp. 263-275.
20. MªE. Manso, M. Genero, and M. Piattini, "No-Redundant Metrics for UML Class Diagrams Structural Complexity", CAISE 2003, *Lecture Notes in Computer Science*, 2681, Springer, Eder, J. and Missikoff, M. (eds.) Springer-Verlag, 2003, pp. 127-142.
21. M. Genero, J. Olivas, M. Piattini, and F. Romero, "Assessing Object Oriented Conceptual Models Maintainability", In Olive et al. (Eds.), International Workshop on Conceptual Modeling Quality (IWCMQ'02), Tampere, Finland, *Lecture Notes in Computer Science*, 2784, Springer-Verlag, 2003, pp. 288-299.
22. Wohlin, C., P. Runeson, M. Höst, M. Ohlson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering: An Introduction*, Kluwer Academic Publishers, 2000.
23. N. Chapin, J. Hale, K. Khan, J. Ramil, and W. Tan, "Types of software evolution and software maintenance", *Journal of Software maintenance and evolution: research and practice*, 13, 2001, pp. 3-30.
24. SPSS 11.0., *Syntax Reference Guide*, Chicago, SPSS Inc., 2001.
25. Kleinbaum D., L. Kupper, and K. Muller, *Applied Regression Analysis and other Multivariate Method,*. Duxbury Press, 1987.
26. Bovas A., and J. Ledolfer, *Statistical Methods for Forecasting*. Wiley Series in Probability and Mathematical Statistics, 1983.
27. Conte S., H. Dunsmore, and V. Shen, *Software Enginering Metrics and Models*. Benjamin-Cummings Publishing Co., Inc. Redwood City, CA, USA, 1986.