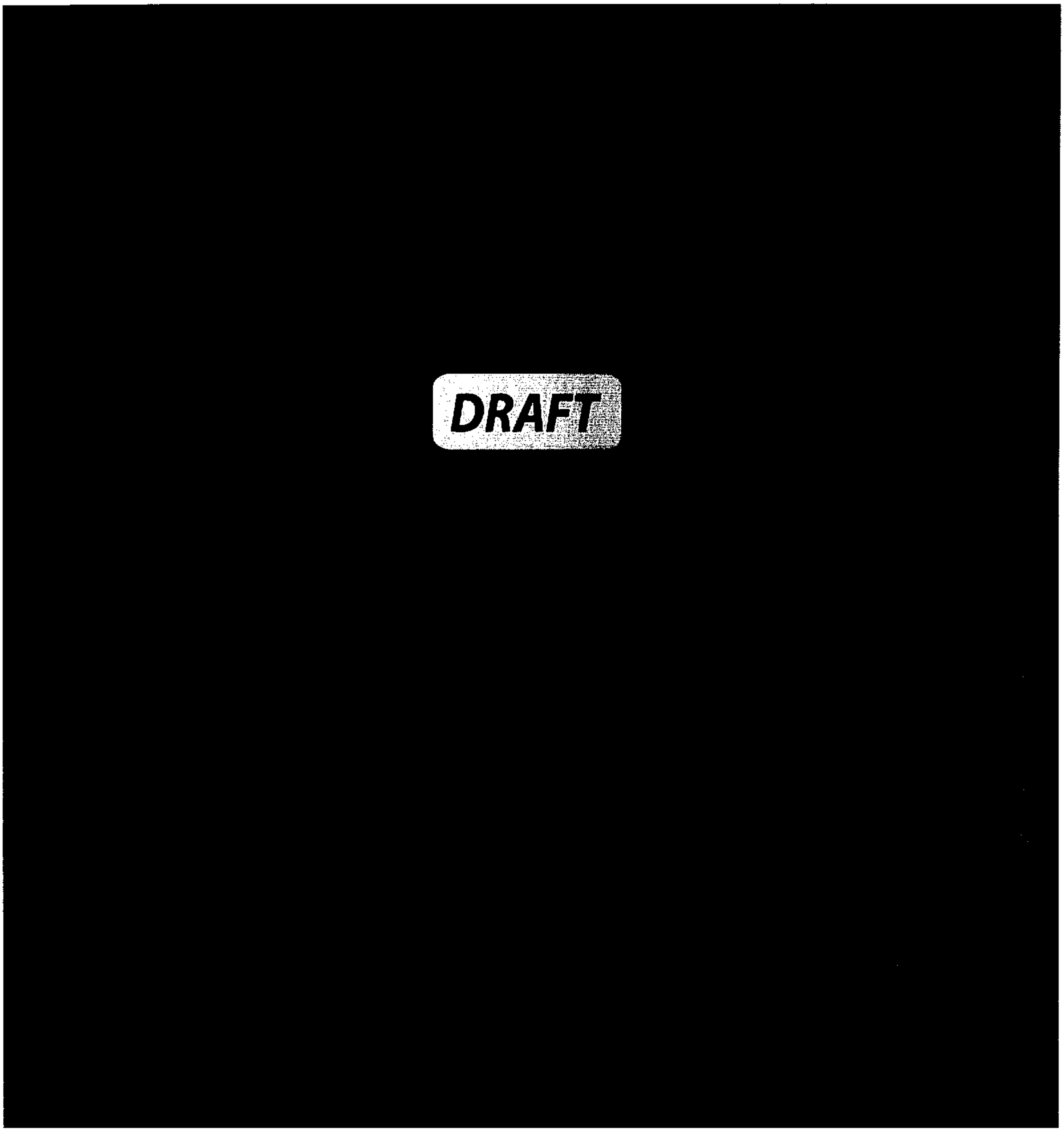
Research Reports in Software Engineering and Management 2008:02

Proceedings of the 3rd Workshop on Quality in Modeling

Jean-Louis Sourrouille Miroslaw Staron (Eds.)

IT University of Göteborg
CHALMERS GÖTEBORGS UNIVERSITET
Department of Applied IT



Research reports in Software Engineering and Management
Report number 2008:02
Series editor: Lars Pareto

ISSN: 1654-4870

Department of Applied Information Technology

IT University of Göteborg

Göteborg University and Chalmers University of Technology

PO Box 8718

SE – 402 75 Göteborg

Sweden

Telephone + 46 (0)31-772 4895

Proceedings of the 3rd

Workshop on Quality in Modeling

Co-located with

MoDELS 2008

The ACM/IEEE 11th International Conference on

Model Driven Engineering Languages and Systems

Editors

Jean-Louis Sourrouille
Miroslaw Staron

Organizers

Jean-Louis Sourrouille, chair,
INSA Lyon, France

Miroslaw Staron, program chair,
IT University of Göteborg, Sweden

Ludwik Kuzniarz,
Blekinge Institute of Technology, Ronneby, Sweden

Parastoo Mohagheghi,
SINTEF ICT, Norway

Lars Pareto,
IT University of Göteborg, Sweden

Program committee

Colin Atkinson, TU Braunschweig , Germany
Thomas Baar, akquinet tech@spree, Berlin, Germany
Benoit Baudry, IRISA-INRIA Rennes, France
Michel Chaudron, Leiden University, The Netherlands
Alexander Förster, University of Paderborn, Germany
Brian Henderson-Sellers, UT Sydney, Australia
Mieczyslaw Kokar, Northeastern University, USA
Kai Koskimies, TU Tampere, Finland
Ludwik Kuzniarz, BTH, Sweden
Christian Lange, Federal Office for Information Technology, Germany
Hervé Leblanc, University of Toulouse, France
Parastoo Mohagheghi, SINTEF ICT, Norway
Lars Pareto, IT University of Göteborg, Sweden
Alexander Pretschner, ETH Zurich, Switzerland
Gianna Reggio, Universita di Genova, Italy
Bernhard Rumpe, TU Braunschweig, Germany
Jean Louis Sourrouille, INSA Lyon, France
Miroslaw Staron, IT University of Göteborg, Sweden
Perdita Stevens, University of Edinburgh, UK

Preface

Quality is an important issue in software engineering, and software managers definitely know the impact of software quality on software production. The recent introduction of Model Driven Software Development raises new challenges, and software quality management is widely researched from multiple perspectives and viewpoints: issues about quality of models need to be approached from both industry practices and academic research.
The workshop is built upon the experience and discussions during the previous workshop on Quality in Modeling. It aims to gather researchers and practitioners interested in the emerging issues of quality in the context of MDD. The workshop is intended to provide a forum for presentation and discussion of emerging issues related to software quality in MDD.

The intention of this year's workshop is to devote a part of the discussion to model quality related to development process. Within "usual" software development, software process quality and project management quality are widely used, while code quality remains an under-exploited way to improve software quality. Not to meet similar problems, a special attention is to pay to practical issues such as the introduction of model quality into the software development process in a convenient and accepted way.

The workshop is divided into two parts:

- Presentation and discussion of the contributions of the accepted paper,
- Working part, guided discussions introduced by industrial practitioners.

The presentation part consists of two sessions for the presentation of accepted paper contributions:

- Towards model quality,
- Frameworks for model quality.

The rationale behind the first session of the working part is to carry out a discussion about the introduction of model quality into software development process by making a parallel with quality of code. First, an industrial practitioner will present practical aspects and solutions regarding code quality in actual software development. Then participants will discuss a list of prepared questions aiming to define practical solutions and delimit pending issues.
The second session of the working part will deal with desirable future works and research interests of the participants, aiming to draw a map of the promising research directions for Quality in Modeling.

The summary and results of the working sessions will be published in the post-workshop report.

Workshop organizers

Table of contents

Empirical Validation of Measures for UML Class Diagrams: A Meta-Analysis Study

M. Esperanza Manso[1], José A. Cruz-Lemus[2], Marcela Genero[2], Mario Piattini[2]

[1] GIRO Research Group, Department of Computer Science, University of Valladolid, Campus Miguel Delibes, E.T.I.C., 47011, Valladolid, Spain
manso@infor.uva.es

2ALARCOS Research Group, Department of Technologies and Information Systems, University of Castilla-La Mancha, Paseo de la Universidad, 4
13071 Ciudad Real, Spain
JoseAntonio.Cruz@uclm.es Marcela.Genero@uclm.e Mario.Piattini@uclm.es

Abstract. The main goal of this paper is to show the findings obtained through a meta-analysis study carried out with the data obtained from a family of controlled experiments. This consisted of 5 experiments performed in academic environments, which were carried out to validate empirically two hypotheses applied to UML class diagrams. These hypotheses investigate 1) The dependence between the structural complexity and size of UML class diagrams on one hand and their cognitive complexity on the other, as well as 2) The dependence between the cognitive complexity of UML class diagrams and their comprehensibility and modifiability. We carried out a meta-analysis, as it allows us to integrate the individual findings obtained from the execution of a family of experiments carried out to test the aforementioned hypotheses. The meta-analysis reveals that the measures related to associations and generalizations have a strong correlation with the cognitive complexity, and that the cognitive complexity has a greater correlation to comprehensibility than to modifiability. These results have implications from the points of view of both modeling and teaching, revealing which UML constructs are most influential when modelers have to comprehend and modify UML class diagrams. In addition, the measures related to associations and generalizations could be used to build prediction models.

Keywords: *meta-analysis, experiments, UML class diagrams, comprehensibility, modifiability, structural complexity, size.*

1. Introduction

The Model-Driven Development paradigm (MDD) [2] is an emerging approach for software development which is of ever-increasing interest to both the research community and software practitioners. MDD considers models as end-products rather than simply as means to produce software. The basic strategy in this approach is the use of model transformations to obtain the final software product. In this context the quality focus has shifted from code to models, given that the quality of the models obtained through transformations is of great importance. This is because it will ultimately determine the quality of the software systems produced. Since, in the context of MDD, maintenance must be done on models, we are concerned about sub-characteristics of maintainability, such as the comprehensibility and modifiability of UML class diagrams. Class diagrams constitute the backbone of a system design and they must be comprehensible and flexible enough to allow the modifications that reflect changes in the things they model to be incorporated easily. We have based our work on the model shown in Figure 1 [1, 3]. This model constitutes a theoretical basis for the development of quantitative models relating to internal and external quality attributes and has been used as the basis for a great amount of empirical research into the area of structural properties of software artefacts [4-6]. In the study reported here, we have assumed a similar representation for UML class diagrams. We hypothesize that the structural properties (such as structural complexity and size) of a UML class diagram have an effect on its cognitive complexity. Cognitive complexity can be defined as the mental burden placed by the artefact on the people who have to deal with it (e.g. modellers, maintainers). High cognitive complexity will result in the production of an artefact that has reduced comprehensibility and modifiability, which will consequently affect its maintainability.

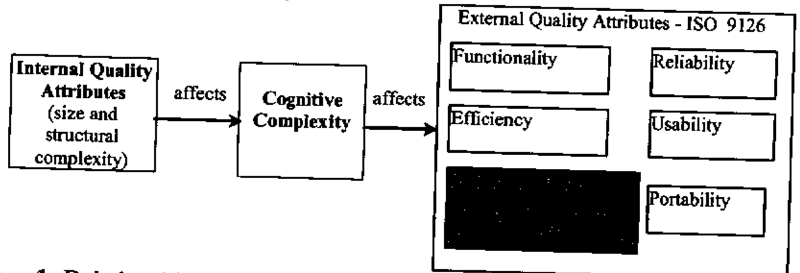


Figure 1. Relationship between structural properties, cognitive complexity, and external quality attributes, based on [1, 3]

The main motivation behind the research we have been carrying out is to validate this model, formulating two main hypotheses based on each of the arrows in Figure 1: 1) Size and structural complexity of UML class diagrams affect cognitive complexity, 2) Cognitive complexity affects the comprehensibility and modifiability of UML class diagrams. To measure the content of each box of Figure 1 we have defined some measures, which will be introduced in Section 3. In order to test such hypotheses, we carried out 5 experiments ,which constitute a family of experiments [7, 8].

The data analysis carried out in each individual experiment did not allow us to obtain conclusive results. This led us to carry out a meta-analysis study. Meta-analysis has been recognised as an appropriate way to aggregate or integrate the findings of empirical studies in order to build a solid body of knowledge on a topic based on empirical evidence [9-11]. Moreover, the need for meta-analysis is gaining relevance in empirical research, as is demonstrated by the fact that it is a recurrent topic in various forums related to Empirical Software Engineering. In other areas, such as psychology or medicine, a single study is extremely unlikely to be definitive. Dozens and even hundreds of studies on the same topic may follow. In Empirical Software Engineering it is unusual for a large amount of studies concerning the same topic to take place, but it is necessary to cross the borders of individual studies so as to extract more global conclusions from families of experiments, with or without significant results. Meta-analysis is a tool for extracting these global conclusions from families of experiments, as it allows us to estimate the global effect size of the whole family, as well as to measure the accuracy of this measure and to evaluate the significance of effect size with respect to the hypotheses under study.

The main goal of the current paper is to present a meta-analysis study that would serve to integrate the results obtained from previous experimentation. In this way, meta-analysis contributes to the obtaining of a solid body of knowledge concerning the usefulness of the measures for UML Class diagrams.

The remainder of the paper is organised as follows: Section 2 presents the related work; Section 3 describes the family of experiments. The Meta-analysis study is presented in Section 4. Finally, the last section presents some concluding remarks and outlines our future work.

2. Related work

In empirical studies within the context of Empirical Software Engineering, special interest has been placed on external quality attributes such as maintainability, comprehensibility, modifiability, etc. Initially, the focus was on code or detailed design artifacts [12-16]. Later, given the increasing relevance of modeling, the focus shifted to models. The comprehension and modification of UML diagrams have been the goals of a great amount of the empirical research on UML diagrams carried out in recent years [17-25].

Our previous works address the influence of both the structural complexity and size of UML class diagrams on their comprehensibility and modifiability. A summary of these works is shown in [8]. In all of them, several controlled experiments were carried out, but the data analysis took place individually for each experiment, in some cases obtaining controversial results. For this reason and owing to the increasing need to investigate the UML constructs that have most influence on the comprehension and modification of UML class diagrams, we decided to integrate the results of homogenous experiments through a meta-analysis study, which is the main goal of the current work.

3. The Family of Experiments

Isolated studies (or experiments) hardly ever provide enough information to answer e questions posed in a research study [10, 26, 27]. Thus, it is important for experi- 'ts to be part of families of studies [26]. Common families of studies can contrib- devising important and relevant hypotheses that may not be suggested by indi- 'xperiments. More importantly, they allow researchers to answer questions that 1d the scope of individual experiments, and to generalize findings across studies. In this work we will comment on five experiments, whose main contextual characteristics are summarized in Table 1.

Table 1. Characteristics of the experiments

Study	#Subjects	University	Date	Year
E1	72	University of Seville (Spain)	March 2003	4th
R1	28		March 2003	
E2	38	Univ. of Castilla-La Mancha (Spain)	April 2003	3rd
R21	23	University of Sannio (Italy)	June 2003	4th
R22	71	University of Valladolid (Spain)	Sept. 2005	3rd

To perform the experiments, we followed the guidelines provided in [28, 29].

3.1 Planning of Experiments

In this sub-section we will define the common framework of all the studies:

1. **Preparation.** The family has a double goal, defined as:
   - Goal 1: To analyze the structural complexity of UML class diagrams with respect to their relationship with cognitive complexity from the viewpoint of software modelers or designers in an academic context.
   - Goal 2: To analyze the cognitive complexity of UML class diagrams with respect to their relationship with comprehensibility and modifiability from the viewpoint of software modelers or designers in an academic context.
2. **Context definition.** In these studies, we have used students as experimental subjects. The tasks to be performed did not require high levels of industrial experience, so we believed that these subjects might be considered appropriate, as is pointed out in several works [26, 30]. In addition, working with students implies a set of advantages, such as the facts that the students' prior knowledge is fairly homogeneous, a large number of subjects is readily available, and there is the possibility of testing experimental design and initial hypotheses [31]. A further advantage of using novices as subjects in experiments on understandability is that the cognitive complexity of the object under study is not hidden by the subjects' experience.
3. **Material.** The experimental materials consisted of a set of UML class diagrams suitable for the family goals. The selected UML class diagrams covered a wide range of the metrics values, considering three types of diagrams: Difficult to maintain (D), Easy to Maintain (E) and Moderately difficult to maintain (M). Some were specifically designed for the experiments and others were obtained from real applications. Each diagram had some documentation attached, containing, among other things, four comprehension and four modification tasks.

studies. In this work we will comment on five experiments, whose main contextual characteristics are summarized in Table 1.

Table 1. Characteristics of the experiments

Study	#Subjects	University	Date	Year
E1	72	University of Seville (Spain)	March 2003	4th
R1	28		March 2003	
E2	38	Univ. of Castilla-La Mancha (Spain)	April 2003	3rd
R21	23	University of Sannio (Italy)	June 2003	4th
R22	71	University of Valladolid (Spain)	Sept. 2005	3rd

To perform the experiments, we followed the guidelines provided in [28, 29].

3.1 Planning of Experiments

In this sub-section we will define the common framework of all the studies:

1. **Preparation.** The family has a double goal, defined as:
   - Goal 1: To analyze the structural complexity of UML class diagrams with respect to their relationship with cognitive complexity from the viewpoint of software modelers or designers in an academic context.
   - Goal 2: To analyze the cognitive complexity of UML class diagrams with respect to their relationship with comprehensibility and modifiability from the viewpoint of software modelers or designers in an academic context.
2. **Context definition.** In these studies, we have used students as experimental subjects. The tasks to be performed did not require high levels of industrial experience, so we believed that these subjects might be considered appropriate, as is pointed out in several works [26, 30]. In addition, working with students implies a set of advantages, such as the facts that the students' prior knowledge is fairly homogeneous, a large number of subjects is readily available, and there is the possibility of testing experimental design and initial hypotheses [31]. A further advantage of using novices as subjects in experiments on understandability is that the cognitive complexity of the object under study is not hidden by the subjects' experience.
3. **Material.** The experimental materials consisted of a set of UML class diagrams suitable for the family goals. The selected UML class diagrams covered a wide range of the metrics values, considering three types of diagrams: Difficult to maintain (D), Easy to Maintain (E) and Moderately difficult to maintain (M). Some were specifically designed for the experiments and others were obtained from real applications. Each diagram had some documentation attached, containing, among other things, four comprehension and four modification tasks.

3.2. How the Individual Experiments were conducted.

We shall now explain the experimental plan of the different members of the family of experiments. The variables considered for measuring the structural complexity and size were the set of 11 measures presented in Table 7 in Appendix A. The *CompSub* measure is the subjective perception given by the subjects with regard to the complexity of the diagrams they have to work with during the experimental task. We consider *CompSub* to be a measure of cognitive complexity. The allowable values of this variable are: Very simple, Moderately simple, Average, Moderately complex and Very complex. To measure the comprehensibility and modifiability of UML class diagrams, we considered the time (in seconds) taken by each subject to complete the comprehensibility and modifiability tasks. We called these measures the *Comprehensibility* and *Modifiability* time.

We used a counter-balanced between-subjects design, i.e., each subject works with only one diagram. The diagrams were randomly assigned and each diagram is considered by the same number of subjects.

We formulated the following hypotheses, which are derived from the family's goals:

- $H_{0,1}$: The structural complexity and size of UML class diagrams are not correlated with the cognitive complexity. $H_{1,1}$: $\neg H_{0,1}$
- $H_{0,2}$: The cognitive complexity of UML class diagrams is not correlated with their comprehensibility and modifiability. $H_{1,2}$: $\neg H_{0,2}$

All the experiments were supervised and time-limited. More details can be found in [7, 8]. Finally, we used SPSS [32] to perform all the statistical analyses and the tool Comprehensive Meta Analysis [33] was employed to perform the meta-analysis.

3.3 Experiment 1 (E1) and Replication (R1)

On testing the hypotheses we obtained the following findings:

- **Correlation between structural and cognitive complexities (hypotheses 1 - goal 1).** The correlation between the *CompSub* variable and the 11 metrics was significant at a 0.05 level for E1. We also obtained a significant correlation for R1 in all cases, with the exception of the NM, NGen and MaxDIT metrics. Because of constraints on space, we do not include the coefficient tables here.
- **Correlation between cognitive complexity and comprehensibility and modifiability (hypotheses 2 - goal 2).** Table 2 indicates that for E1, the subjective complexity is significantly correlated to the comprehensibility. For R1, the results are not significant, and they are hence unfavourable to goal 2 of the family.

Table 2. Results related to goal 2 for E1 & R1

Variables correlated	E1 (n=62)		R1(n=22)	
	$\rho_{spearman}$	p-value	$\rho_{spearman}$	p-value
CompSub vs *Comprehensibility*	0.266	0.037	0.348	0,111
CompSub vs *Modifiability*	0.132	0.306	0.270	0.217

The results obtained are now related to the family's goals:

- Goal 1: Structural and cognitive complexities present a positive significant correlation for all metrics, except for NM, NGen and MaxDIT in R1.
- Goal 2: Cognitive complexity seems to be positively correlated to the effort needed to comprehend UML class diagrams, but the results are significant only for E1. At the same time, there is no correlation with the effort needed to modify the diagrams. A possible explanation for this could be that the subjects base their perception on the difficulty of the first tasks that they perform, which in this case are the comprehension ones.

3.4 Experiment 2 (E2) and its Replications (R21 & R22)

In these studies, goals and variables are the same as in the previous ones, but the diagrams used were different, and context and design have also been improved. More detailed information about them can be found in [8].

Apart from the family's variables, some other variables have been added, in order to validate the results:

- CompCorrectness = # correct comprehension tasks / # total tasks performed
- CompCompleteness = # correct comprehension tasks / # total tasks to perform
- ModifCorrectness = # correct modification tasks / # total tasks performed
- ModifCompleteness = # correct modification tasks / # total tasks to perform

Again, we use a within-subjects design, but in this case it has been improved by blocking the subjects' experience. A pre-test was performed, the results of which led to the subjects' being divided into two groups. Each diagram was then assigned to the same number of subjects from each group. More details about this process can be found in [8].

The *Comprehensibility* and *Modifiability* measures were only included when the tasks performed had a minimum quality level, and it was for this reason that we used the newly introduced variables, presented previously. The subjects who attained under 75% in correctness and completeness were excluded from the study. In fact their exclusion improved the behaviour of the dependent variables, i.e, symmetry and outliers.

On testing the hypotheses we obtained the following findings:

- **Correlation between** structural **and cognitive complexities (hypothesis 1-goal 1).** Table 3 summarizes the metrics that are significantly correlated with the *CompSub* variable.

Table 3. Goal 1 results for E2, R21 & R22

Study	Significantly correlated metrics
E2	NC, NAssoc, NGen, NGenH, MaxDIT (5 out of 11)
R21	All except for NM, NGenH and MaxAgg (8 out of 11)
R22	All except for NM (10 out of 11)

- **Correlation between cognitive complexity and comprehensibility and modifiability (goal 2).** Table 4 indicates that for all the studies, the subjective complexity is significantly correlated to the comprehensibility. For modifiability, the results are not significant in all cases, and are once again unfavourable to goal 2 of the family.

Table 4. Results related to goal 2 for E2, R21 & R22

Variables correlated	E2			R21			R22		
	$\rho_{Spearman}$	p-value	N	$\rho_{Spearman}$	p-value	N	$\rho_{Spearman}$	p-value	N
CompSub vs *Comprehensibility*	0.343	**0.049**	33	0.410	**0.065**	21	0.353	**0.003**	70
CompSub vs *Modifiability*	0.337	0.099	25	0.156	0.500		0.165	0.173	

These studies were based on two goals related to the connection of the different elements of the theoretical model which we used as a starting point. The results were:

- Goal 1: We have favourable results which admit a correlation between the structural and the cognitive complexities of UML class diagrams. Most of the metrics are significantly correlated with the subjective complexity in the different studies, especially those related to inheritance hierarchies.
- Goal 2: The results are also in favour of the hypothesis that relates cognitive complexity to the comprehensibility of UML class diagrams.

3.5 Threats to the Validity of the Family of Experiments.

The main threats to the validity of the family are the following:

- **Conclusions validity.** The number of subjects in R1, E2 and R21 is quite low, and subjects were selected by convenience. Our conclusions must therefore be applied to the population represented by the subjects.
- **Internal validity.** We have found correlation between the variables, which implies the possibility of the existence of that causality, but not the causality itself. Moreover, R21 materials were written in English, which is not the mother language of the subjects (Italians). This fact may have increased the times taken to perform the tasks, especially those of modification.
- **External validity.** It would be advisable to perform some replications with data extracted from real projects, in an effort to generalise the results obtained.

4. Meta-Analysis Study

There are several statistical methods that allow us to accumulate and interpret a set of results obtained through different inter-related experiments, since they check similar hypotheses [34-38]. The limitations of all these methods are commented upon in [11]. There are three main ways in which to perform this process: meta-analysis, significance level combination and vote counting.

The first and second are those most commonly used in Software Engineering, and it is for this reason that we comment upon them now:

- Meta-analysis is a set of statistical techniques that allow us to combine the different effect size measures (or treatment effect) of the individual experiments. There are several metrics to obtain this value, e.g. the means difference and the correlation coefficients, among others [35]. The objective is to obtain a global effect, the treatment effect of all experiments. As effect size measures may come from different environments and may even not be homogeneous, it is necessary to obtain a standardized measure of each one. For example, the dependence between two variables could be measured by different coefficients or scales The global effect size is obtained as a weighted average of standardized measures, in which the most commonly used weights are the sample size or the standard deviation. Together with the estimation of the global effect size, we can provide an estimated confidence interval and a p-value which allows us to decide on the meta-analysis hypotheses. We can find several applications of this technique in Empirical Software Engineering in the following works ,among others:
  - Miller and McDonald [39] synthesize the results of a study on the effect that a tool had on the effectiveness and efficiency of inspections. The effect sizes were obtained from the correlation coefficients.
  - Dybå et al. [40] perform a meta-analysis for studying the effect of pair-programming on quality, duration and effort. The effect size is measured with a mean difference. This technique was also used in [41-43] to obtain conclusions about experiments which evaluated different inspection techniques.
- Signification level combination. A mean, or another statistic is obtained, in order to summarize the different signification levels (α) of the experiments. An application of this technique can be found in [41], in which certain inspection techniques are studied in an experiment and four replications.

In the present study, we have a family of experiments whose main goals are:

1. To study the influence of metrics on the cognitive complexity of UML class diagrams.
2. To study the influence of cognitive complexity on the comprehensibility and modifiability of UML class diagrams.

The use of meta-analysis will allow us to extract global conclusions, despite the fact that some of the experimental conditions are not the same. As we have mentioned previously, we will need to standardize the effect sizes. In this meta-analysis we used correlation coefficients (r_i) that, once transformed (Fisher transformation), provide the effect sizes that have a Normal distribution (z_i), what makes them easier to use. The global effect size is obtained using the Hedges' g metric [35, 44], that is a weighted mean which has the proportional weights to the experiment size (equation 1).

$$\overline{Z} = \frac{\sum_i w_i z_i}{\sum_i w_i} \qquad W_i = 1/(n_i-3) \tag{1}$$

The higher the value of Hedges' g is, the higher the corresponding correlation coefficient is too. For studies in Software Engineering, we can classify effect sizes into small, medium and large [44]. We rely on the use of the five empirical studies, previously presented in this work, which means that the conclusions about our goals will be extracted from five different results.

4.1. Meta-analysis Results

Firstly, a meta-analysis for each metric-*CompSub* pair will be carried out, taking into account the fact that the hypothesis test is one-tailed, i.e., we consider as null-hypothesis that the correlation is now above zero. In Table 5 we present the global estimation of the correlation coefficient, a confidence interval at 95%, the p-value and the value for Hedges' g, including a classification of the effect size as large (L), medium (M) or small (S).

Table 5. Meta-analysis of metrics-*CompSub*

H0: $\rho \leq 0$	Correlation (ρ) Global effect size	Lower limit	Upper limit	p-value	Hedges'g
NC	0.566	0.464	0.653	0.0000	1.322(L)
NA	0.541	0.435	0.632	0.000	1.219(L)
NM	0.177	0.040	0.307	0.012	0.339(S)
Nassoc	0.566	0.465	0.653	0.000	1.318(L)
NAgg	0.481	0.368	0.581	0.000	1.051(M)
NDep	0.484	0.371	0.584	0.000	1.060(M)
NGen	0.484	0.371	0.584	0.000	1.018 (L)
NGenH	0.422	0.302	0.529	0.000	0.903 (M)
NAggH	0.393	0.270	0.504	0.000	0.814 (M)
MaxDIT	0.492	0.379	0.590	0.000	1.080 (L)
MaxHAgg	0.360	0.233	0.474	0.000	0.734 (M)

The results observed are in favour of the existence of a positive correlation between cognitive complexity and the 11 metrics that measure the structural complexity and size of UML class diagrams. In fact, most of the effect sizes are medium or large, with the exception of NM, which is small. The size metrics that have most influence upon the cognitive complexity are NC and NA, while the complexity metrics that have most influence upon cognitive complexity are related to aggregations (NAgg) and generalizations (NGen and MaxDIT). We can conclude that those diagrams with many classes and attributes will have an increased cognitive complexity. Moreover, class diagram models using many inheritance and aggregation mechanisms will also have an increased cognitive complexity.

With regard to the hypotheses derived from goal 2, Table 6 shows that we can admit the existence of correlation between the cognitive complexity and the two measures, *Comprehensibility* and *Modifiability*, which measure quality attributes of UML class diagrams.

Table 6. Meta-analysis of *CompSub-Comprehensibility* and *Modifiability* time

H0: ρ≤0	Correlation (ρ) gobal effect size	Lower limit	Upper limit	p-value	Hedges'g
Comprehensibility Time	0.330	0.200	0.449	0.000	0.684 (M)
Modifiability Time	0.186	0.044	0.320	0.011	0.368(M)

The effect sizes are medium in both cases, but the correlation estimation of *Comprehensibility* is larger than the correlation of *Modifiability*. So we can conclude that, the more cognitive complexity a diagram contains, the more difficult it will be to comprehend and modify.

As an example, Figure 2 presents in diagram form the meta-analysis of the relationship of a couple of metrics and the *CompSub* measure, and the relationship between their comprehensibility and cognitive complexity.

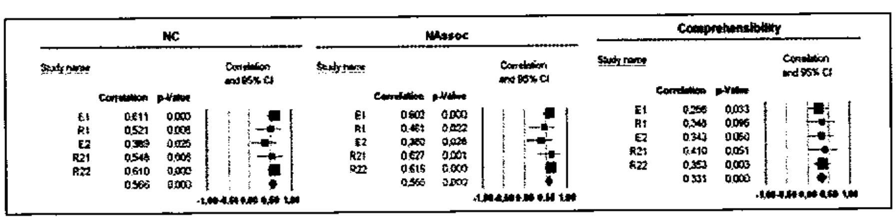


Figure 2. Meta-analysis for NC-*CompSub*, NAssoc-*CompSub* and *CompSub-Comprehensibility*

5. Conclusions

The main goal of this work has been that of validating a theoretical model which relates the structural complexity and size of UML class diagrams and cognitive complexity to two of their external quality attributes: comprehensibility and modifiability

(Figure 1). For that purpose, we carried out a meta-analysis study with the data obtained from a family of five experiments. The meta-analysis results are in favour of the model under inspection with regard to the two goals being pursued:

- Goal 1: structural complexity is correlated with cognitive complexity, especially with that related to associations and generalizations. An increase in the number of classes and attributes within classes also increases the cognitive complexity of UML class diagrams.
- Goal 2: cognitive complexity influences both the comprehensibility time and modifiability time of UML class diagrams, but this is especially true in the former case.

These results are relevant, as they point to a means of controlling the level of certain quality attributes of UML class diagrams from the modelling phase. The findings also have implications, both practically and in terms of teaching, providing information about which UML constructs may have more implications in the effort to understand and maintain UML class diagrams. When alternative designs of UML class diagrams exist, it could be advisable to select the one which minimizes these constructs.

Moreover, the measures related to associations and generalizations could be used to build prediction models, to evaluate how the time taken to understand or modify an UML class diagram increases; we have done this prediction modelling in [8]. In future work we plan to refine the prediction models obtained, using the data obtained in the whole family of experiments.

Further work is needed to confirm the findings of the current study, improving different issues:

1. Increasing the class diagram sample, as the results generalization depends on the sample of objects examined with respect to the whole object population. This meta-analysis covers 33 (24+9) UML class diagrams.
2. Working with subjects from different fields, preferably real practitioners or students from other universities, in the quest to generalize the results with regard to the subject population.
3. Improving the modifying tasks, to make them as real as possible.
4. Investigating other metrics to do with cognitive complexity.

Also pending is the carrying out of a similar study with the measures we have defined for UML statechart diagrams [45] and OCL expressions [46]

Acknowledgements

This research is part of the ESFINGE project (TIN2006-15175-C05-05) financed by the "Ministerio de Educación y Ciencia (Spain)", the IDONEO project (PAC08-0160-6141) financed by "Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha".

References

1. L. Briand, S. Morasca and V. Basili, "Defining and Validating Measures for Object-Based High-Level Design", *IEEE Transactions on Software Engineering*, 25, 1999, pp. 722-743.
2. C. Atkinson and T. Kühne, "Model Driven Development: a Metamodeling Foundation", *IEEE Software*, 20, 2003, pp. 36-41.
3. ISO-IEC, "ISO/IEC 9126. Information Technology - Software Product Quality", 2001.
4. K. El-Emam and W. Melo, "The Prediction of Faulty Classes Using Object-Oriented Design Metrics", National Research Council of Canada, NRC/ERB1064, 1999.
5. K. El-Emam, S. Benlarbi, N. Goel and S. Rai, "The Confounding Effect of Class Size on the Validity of Object-Oriented Metrics", *IEEE Transactions on Software Engineering*, 27(7), 2001, pp. 630-650.
6. G. Poels and G. Dedene, "Measures for Assessing Dynamic Complexity Aspects of Object-Oriented Conceptual Schemes", *19th International Conference on Conceptual Modelling (ER 2000)*, 2000, pp. 499-512.
7. M. Genero, M. E. Manso and M. Piattini, "Early Indicators of UML Class Diagrams Understandability and Modifiability", *ACM-IEEE International Symposium on Empirical Software Engineering*, 2004, pp. 207-216.
8. M. Genero, M. E. Manso, A. Visaggio, G. Canfora and M. Piattini, "Building Measure-Based Prediction Models for UML Class Diagram Maintainability", *Empirical Software Engineering*, 12, 2007, pp. 517-549.
9. M. Lipsey and D. Wison, *Practical Meta-Analysis*, Sage, 2001.
10. J. Miller, "Applying Meta-Analytical Procedures to Software Engineering Experiments", *Journal of Systems and Software*, 54, 2000, pp. 29-39.
11. L. M. Pickard, "Combining Empirical Results in Software Engineering", University of Keele, T-R V1, 2004.
12. W. Li and S. Henry, "Object-Oriented Metrics that Predict Maintainability", *Journal of Systems and Software*, 23, 1993, pp. 111-122.
13. 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.
14. F. Fioravanti and P. Nesi, "Estimation and Prediction Metrics for Adaptative Maintenance Effort of Object-Oriented Systems", *IEEE Transactions on Software Engineering*, 27(12), 2001, pp. 1062-1083.
15. L. Briand, C. Bunse and J. Daly, "A Controlled Experiment for Evaluating Quality Guidelines on the Maintainability of Object-Oriented Designs", *IEEE Transactions on Software Engineering*, 27(6), 2001, pp. 513-530.
16. E. Arisholm and D. I. K. Sjøberg, "Evaluating the Effect of a Delegated versus Centralized Control Style on the Maintainability of Object-Oriented Software", *IEEE Transactions on Software Engineering*, 30(8), 2004, pp. 521-534.
17. M. C. Otero and J. J. Dolado, "Evaluation of the Comprehension of the Dynamic Modeling in UML", *Information and Software Technology*, 46(1), 2004, pp. 35-53.
18. M. C. Otero and J. J. Dolado, "An Empirical Comparison of the Dynamic Modeling in OML and UML", *Journal of Systems and Software*, 77(2), 2005, pp. 91-102.