## Lecture Notes in Computer Science

The LNCS series reports state-of-the-art results in computer science research, development, and education, at a high level and in both printed and electronic form. Enjoying tight cooperation with the R&D community, with numerous individuals, as well as with prestigious organizations and societies, LNCS has grown into the most comprehensive computer science research forum available.

The scope of LNCS, including its subseries LNAI and LNBI, spans the whole range of computer science and information technology including interdisciplinary topics in a variety of application fields. The type of material published traditionally includes

- proceedings (published in time for the respective conference)
- post-proceedings (consisting of thoroughly revised final full papers)
- research monographs (which may be based on outstanding PhD work, research projects, technical reports, etc.)

[illegible] have been added featuring, [illegible] value components; these [illegible]

[illegible] of lectures given at [illegible]

[illegible] and mediated coverage [illegible]

[illegible] the broader community)

[illegible] is published electronically [illegible]

[illegible]

[illegible] Germany

[illegible]

ISSN [illegible]

ISBN 978-3-642-01647-9

9 783642 016479

springer.com

**Lecture Notes in Computer Science**

LNCS LNAI LNBI

Chaudron (Ed.)

LNCS 5421

Models in Software Engineering

MODE[illegible]
2008
Worksh[illegible]

LNCS 5421

[illegible] Chaud[illegible]

# [illegible]dels in [illegible]tware Eng[illegible]g

### Workshops and Symposia at MODELS 2008
### Toulouse, France, September/October 2008
### Reports and Revised Selected Papers

Springer

Michel R.V. Chaudron (Ed.)

# Models in Software Engineering

Workshops and Symposia at MODELS 2008
Toulouse, France, September 28 – October 3, 2008
Reports and Revised Selected Papers

Springer

Volume Editor

Michel R.V. Chaudron
University of Leiden
Faculty of Science
Leiden Institute of Advanced Computer Science
P.O. Box 9512, 2300 RA Leiden, The Netherlands
E-mail: chaudron@liacs.nl

Library of Congress Control Number: Applied for

CR Subject Classification (1998): D.2, D.3, I.6, K.6

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-642-01647-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-01647-9 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12607341 06/3180 5 4 3 2 1 0

# Preface

Following the tradition of previous editions of the MODELS conference, many satellite events were organized in co-location with the MODELS conference in Toulouse in 2008: 12 workshops, 3 symposia, 9 tutorials, a poster session, and a tools exhibition. The selection of the workshops was organized by a Workshop Selection Committee, which consisted of the following experts:

- Michel R.V. Chaudron, Leiden University, The Netherlands (Chair)
- Jochen Küster, IBM Research Zurich, Switzerland
- Henry Muccini, University of L'Aquila, Italy
- Holger Giese, Hasso-Plattner-Institute, Germany
- Hans Vangheluwe, McGill University, Canada

Some workshops have been running for several years as MODELS satellite events, but each year some workshops end. Furthermore, there are always new developments, and hence there is room for new workshops. Therefore, the Workshop Selection Committee very much welcomes new proposals.

The workshops enabled groups of participants to exchange recent and/or preliminary results, to conduct intensive discussions, or to coordinate efforts between representatives of a technical community. They served as forums for lively discussion of innovative ideas, recent progress, or practical experience on model-driven engineering for specific aspects, specific problems, or domain-specific needs.

The three symposia this year were: the Doctoral Symposium, the Educators' Symposium, and the Research Projects Symposium. The Doctoral Symposium provided specific support for PhD students to discuss their work and receive guidance for the completion of their dissertation research. The Educators' Symposium addressed the question of how to educate students and practitioners to move from traditional thinking to an engineering approach based on models. The Research Projects Symposium was a showcase for research projects, as well as a forum where researchers from academia and industry and representatives of funding agencies could debate on technology transfer and trends in research projects.

These satellite-event proceedings were published after the conference and include summaries as well as revised versions of the best papers from the workshops, the Doctoral Symposium, the Educators' Symposium, and the Research Projects Symposium.

I would like to thank everyone involved in making the satellite events such a successful and inspiring experience.

January 2009

Michel R.V. Chaudron

# Table of Contents

## Model Based Architecting and Construction of Embedded Systems (ACES-MB)

## Challenges in Model Driven Software Engineering (CHAMDE)

## Empirical Studies of Model Driven Engineering (ESMDA)

## Models@runtime

## Model Co-evolution and Consistency Management (MCCM)

## Model-Driven Web Engineering (MDWE)

## Modeling Security (MODSEC)

## Model-Based Design of Trustworthy Health Information Systems (MOTHIS)

## Non-Functional System Properties in Domain Specific Modeling Languages (NFPin DSML)

## OCL Tools: From Implementation to Evaluation and Comparison (OCL)

## Quality in Modeling (QIM)

## Transforming and Weaving Ontologies and Model Driven Engineering (TWOMDE)

## Doctoral Symposium

## Educational Symposium

## Research Project Symposium

aspect needed to help increasing models' "good properties": it detects all rules violations but also provides hints, warns to avoid potential errors, and may include company know-how. Finally, a style guide is a quite necessary complement to put into practice quality assessment.

## References

1. Ambler, S.W.: The Elements of UML 2.0 Style. Cambridge University Press, Cambridge (2005)
2. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide. Addison-Wesley, Reading (1998)
3. Bottoni, P., Koch, M., Parisi-Presicce, F., Taentzer, G.: A visualization of OCL using collaborations. In: Gogolla, M., Kobryn, C. (eds.) UML 2001. LNCS, vol. 2185, pp. 257–271. Springer, Heidelberg (2001)
4. Caplat, G., Sourrouille, J.L.: MDA: Model Mapping using Formalism Extension. IEEE Software 22(2), 44–51 (2005)
5. Farkas, T., Hein, C., Ritter, T.: Automatic Evaluation of Modeling Rules and Design Guidelines. In: Proc. of the Workshop From code centric to Model centric Soft. Eng., `http://www.esi.es/modelware/c2m/papers.php`
6. Harel, D., Rumpe, B.: Modeling Languages: Syntax, Semantics and All That Stuff, TR MCS00-16, The Weizmann Institute of Science (2000)
7. Hnatkowska, B.: Verification of Good Design Style of UML Models. In: Proc. Int. Conf. Information System Implementation and Modeling (2007)
8. ISO, International Organization for Standardization, ISO 9126-1:2001, Software engineering – Product quality, Part 1: Quality model (2001)
9. Jacobson, I., Booch, G., Rumbaugh, J.: The Unified Software Development Process. Addison-Wesley, Reading (1999)
10. Kiesner, C., Taentzer, G., Winkelmann, J.: Visual OCL: A Visual Notation of the Object Constraint Language. Technical Report 2002/23, Tech. Univ. of Berlin (2002)
11. Kuzniarz, L., Pareto, L., Sourrouille, J.-L., Staron, M.: Third intenational workshop on quality in modeling. In: Giese, H. (ed.) MoDELS 2007 Workshops. LNCS, vol. 5002, pp. 271–274. Springer, Heidelberg (2008)
12. Lange, C.F.J., DuBois, B., Chaudron, M.R.V., Demeyer, S.: Experimentally investigating the effectiveness and effort of modeling conventions for the UML. CS-Report 06-14, Tech. Univ. Eindhoven (2006)
13. Malgouyres, H., Motet, G.: A UML model consistency verification approach based on meta-modelling formalization. In: Biham, E., Youssef, A.M. (eds.) SAC 2006. LNCS, vol. 4356, pp. 1804–1809. Springer, Heidelberg (2007)
14. Sourrouille, J.-L., Caplat, G.: A Pragmatic View about Consistency Checking of UML Model, Work. Consistency Problems in UML-Based Software Dev., pp. 43–50 (2003)
15. UML, OMG Unified Modeling Language, Version 2.1.2 (2007)
16. Sourrouille, J.L., Hindawi, M., Morel, L., Aubry, R.: Specifying consistent subsets of UML. In: Proc. Educators Symposium, Models 2008, pp. 26–38 (2008) ISBN 83-916444-8-0
17. Hindawi, M., Morel, L., Aubry, R., Sourrouille, J.L.: Description and Implementation of a Style Guide for UML. In: Proc. 3rd Workshop on Quality in Modeling, IT Univ Göteborg RR 2008:02, pp. 31–45 (2008) ISSN 1654-4870

---

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

M. Esperanza Manso[1], José A. Cruz-Lemus[2], Marcela Genero[2], and 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`
[2] ALARCOS Research Group, Department of Technologies and Information Systems,
University of Castilla-La Mancha, Paseo de la Universidad, 4, 13071 Ciudad Real, Spain
`{JoseAntonio.Cruz,Marcela.Genero,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 five controlled experiments performed in academic environments. This family of experiments was carried out to validate empirically two hypotheses applied to UML class diagrams, which 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) [1] 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. 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

M.R.V. Chaudron (Ed.): MODELS 2008 Workshops, LNCS 5421, pp. 303–313, 2009.

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 [2, 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.

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:

Size and structural complexity of UML class diagrams affect cognitive complexity

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. 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.

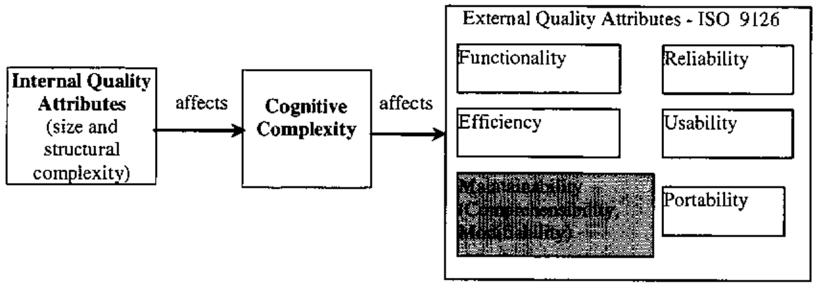

**Fig. 1.** Relationship between structural properties, cognitive complexity, and external quality attributes, based on [1, 3]

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 describes the family of experiments. The Meta-analysis study is presented in Section 3. Finally, the last section presents some concluding remarks and outlines our future work.

## 2 The Family of Experiments

Isolated studies (or experiments) hardly ever provide enough information to answer the questions posed in a research study [10, 12, 13]. Thus, it is important for experiments to be part of families of studies [12]. Common families of studies allow researchers to answer questions that are beyond 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 [14, 15].

### 2.1 Planning of Experiments

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

1. **Preparation**. The family has the goal of testing both the hypotheses presented in the introduction.
    - 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.
    - 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 [12, 16]. 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 [17]. 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.

### 2.2 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 [18] to perform all the statistical analyses and the tool Comprehensive Meta Analysis [19] was employed to perform the meta-analysis.

### 2.3 Experiment 1 (E1) and Replication (R1)

On testing the hypotheses we obtained the following findings:

- 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.
- The subjective complexity seems to be positively correlated to the effort needed to comprehend UML class diagrams, but the results are significant only for E1 (see Table 2). 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.

**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 |

### 2.4 Experiment 2 (E2) and its Replications (R21 and 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 between-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:

**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) |

**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 | |

- 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 (see Table 3).
- The results are also in favour of the hypothesis that relates cognitive complexity to the comprehensibility of UML class diagrams (see Table 4).

### 2.5 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 these 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.

## 3 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 [20-24]. There are three main ways in which to perform this process: meta-analysis, significance level combination and vote counting.

According with the characteristic of our data, in the present study we used meta-analysis, which 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 [21]. 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 [25, 26].

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 [21, 27], 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 [27]. 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.

### 3.1 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).

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.

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.

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

| $H_0$: ρ≤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) |

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

| $H_0$: 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) |

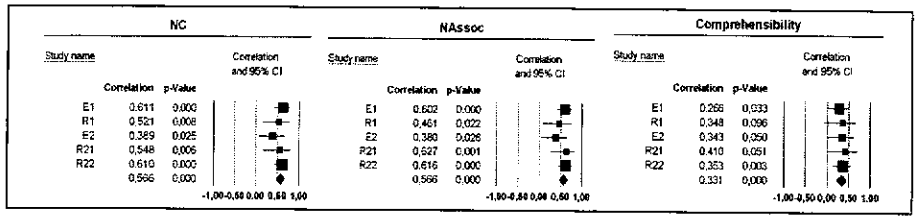

**Fig. 2.** Meta-analysis for NC-*CompSub*, NAssoc-*CompSub* and *CompSub-Comprehensibility*

## 4 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 modeling in [8]. In future work we plan to refine the prediction models obtained, using the data obtained in the whole family of experiments.

Further experimentation is needed to confirm the findings of the current study, improving different issues: 1) Increasing the class diagram sample, 2) Working with practitioners, 3) Improving the modifying tasks and 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 [28] and OCL expressions [29].

## 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. Atkinson, C., Kühne, T.: Model Driven Development: a Metamodeling Foundation. IEEE Transactions on Software Engineering 20, 36–41 (2003)
2. Briand, L., Morasca, S., Basili, V.: Defining and Validating Measures for Object-Based High-Level Design. IEEE Transactions on Software Engineering 25, 722–743 (1999)
3. ISO-IEC, ISO/IEC 9126. Information Technology - Software Product Quality (2001)
4. El-Emam, K., Melo, W.: The Prediction of Faulty Classes Using Object-Oriented Design Metrics, National Research Council of Canada (1999)
5. El-Emam, K., Benlarbi, S., Goel, N., Rai, S.: The Confounding Effect of Class Size on the Validity of Object-Oriented Metrics. IEEE Transactions on Software Engineering 27(7), 630–650 (2001)
6. Poels, G., Dedene, G.: Measures for assessing dynamic complexity aspects of object-oriented conceptual schemes. In: Laender, A.H.F., Liddle, S.W., Storey, V.C. (eds.) ER 2000. LNCS, vol. 1920, pp. 499–512. Springer, Heidelberg (2000)
7. Genero, M., Manso, M.E., Piattini, M.: Early Indicators of UML Class Diagrams Understandability and Modifiability. In: ACM-IEEE International Symposium on Empirical Software Engineering (2004)
8. Genero, M., Manso, M.E., Visaggio, A., Canfora, G., Piattini, M.: Building Measure-Based Prediction Models for UML Class Diagram Maintainability. Empirical Software Engineering 12, 517–549 (2007)
9. Lipsey, M., Wison, D.: Practical Meta-Analysis. Sage, Thousand Oaks (2001)

10. Miller, J.: Applying Meta-Analytical Procedures to Software Engineering Experiments. Journal of Systems and Software 54, 29–39 (2000)
11. Pickard, L.M.: Combining Empirical Results in Software Engineering, University of Keele (2004)
12. Basili, V., Shull, F., Lanubile, F.: Building Knowledge through Families of Experiments. IEEE Transactions on Software Engineering 25, 456–473 (1999)
13. Shull, F., Carver, J., Travassos, G., Maldodano, J., Conradi, R., Basili, V.: Replicated Studies: Building a Body of Knowledge about Software Reading Techniques. Lecture Notes on Empirical Software Engineering, pp. 39–84. World Scientific, Singapore (2003)
14. Wohlin, C., Runeson, P., Hast, M., Ohlsson, M.C., Regnell, B., Wesslen, A.: Experimentation in Software Engineering: an Introduction. Kluwer Academic Publisher, Dordrecht (2000)
15. Juristo, N., Moreno, A.: Basics of Software Engineering Experimentation. Kluwer Academic Publishers, Dordrecht (2001)
16. Höst, M., Regnell, B., Wohlin, C.: Using Students as Subjects - a Comparative Study of Students & Professionals in Lead-Time Impact Assessment. In: 4th Conference on Empirical Assessment & Evaluation in Software Engineering (EASE 2000). Keele, UK (2000)
17. Sjoberg, D.I.K., Hannay, J.E., Hansen, O., Kampenes, V., Karahasanovic, A., Liborg, N.K., Rekdal, A.C.: A Survey of Controlled Experiments in Software Engineering. IEEE Transactions on Software Engineering 31(9), 733–753 (2005)
18. SPSS, SPSS 12.0, Syntax Reference Guide. SPSS Inc.: Chicago, USA (2003)
19. Biostat, Comprehensive Meta-Analysis v2 (2006)
20. Glass, G.V., McGaw, B., Smith, M.L.: Meta-Analysis in Social Research. Sage Publications, Thousand Oaks (1981)
21. Hedges, L.V., Olkin, I.: Statistical Methods for Meta-Analysis. Academia Press (1985)
22. Rosenthal, R.: Meta-Analytic Procedures for Social Research. Sage Publications, Thousand Oaks (1986)
23. Sutton, J.A., Abrams, R.K., Jones, R.D., Sheldon, A.T., Song, F.: Methods for Meta-Analysis in Medical Research. John Wiley & Sons, Chichester (2001)
24. Wolf, F.M.: Meta-Analysis: Quantitative Methods for Research Synthesis. Sage Publications, Thousand Oaks (1986)
25. Dybå, T., Arisholm, E., Sjøberg, D.I.K., Hannay, J.E., Shull, F.: Are Two Heads Better than One? On the Effectiveness of Pair Programming. IEEE Software 24(6), 10–13 (2007)
26. Miller, J., McDonald, F.: Statistical Analysis of Two Experimental Studies, University of Strathclyde (1998)
27. Kampenes, V., Dybå, T., Hannay, J.E., Sjoberg, D.I.K.: A Systematic Review of Effect Size in Software Engineering Experiments. Information and Software Technology 49(11-12), 1073–1086 (2007)
28. Cruz-Lemus, J.A., Genero, M., Piattini, M.: Metrics for UML Statechart Diagrams. In: Genero, M., Piattini, M., Calero, C. (eds.) Metrics for Software Conceptual Models. Imperial College Press, UK (2005)
29. Reynoso, L., Genero, M., Piattini, M.: Measuring OCL Expressions: An approach based on Cognitive Techniques. In: Genero, P.A.C. (ed.) Metrics for Software Conceptual Models. Imperial College Press, UK (2005)
30. Genero, M., Piattini, M., Calero, C.: Early Measures for UML Class Diagrams. L'Object 6(4), 495–515 (2000)
31. Genero, M., Poels, G., Manso, M.E., Piattini, M.: Defining and Validating Metrics for UML Class Diagrams. In: Genero, M., Piattini, M., Calero, C. (eds.) Metrics for Software Conceptual Models. Imperial College Press, UK (2005)

32. Chidamber, S., Kemerer, C.: A Metrics Suite for Object-Oriented Design. IEEE Transactions on Software Engineering 20, 476–493 (1994)

## Appendix A

After studying the UML metamodel, and having reviewed the literature concerning existing measures, we proposed a set of eight measures for the structural complexity of UML class diagrams [30, 31]. The proposed measures are related to the usage of UML relationships, such as associations, dependencies, aggregations and generalizations. In the study reported in this work, we have also considered traditional OO measures, such as size measures (see Table 7).

**Table 7.** Measures for UML class diagrams

| | Measure Name | Measure definition |
|---|---|---|
| Size measures | Number of Classes (NC) | The total number of classes in a class diagram. |
| | Number of Attributes (NA) | The number of attributes defined across all classes in a class diagram (not including inherited attributes or attributes defined within methods). This includes attributes defined at class and instance level. |
| | Number of Methods (NM) | The total number of methods defined across all classes in a class diagram, not including inherited methods (as this would lead to double counting). This includes methods defined at class and instance level. |
| Structural complexity measures | Number of Associations (NAssoc) | The total number of association relationships in a class diagram. |
| | 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 Generalizations (NGen) | The total number of generalization relationships (each "parent-child" pair in a generalization relationship). |
| | Number of Generalization Hierarchies (NGenH) | The total number of generalization hierarchies, i.e. it counts the total number of structures with generalization relationships. |
| | Number of Aggregation Hierarchies (NAggH) | The total number of aggregation hierarchies, i.e. it counts the total numbers of "whole-part" structures within a class diagram. |
| | Maximum DIT (MaxDIT). | The maximum DIT value obtained for each class of the class diagram. The DIT value for a class within a generalization hierarchy is the longest path from the class to the root of the hierarchy [32]. |
| | Maximum HAgg (MaxHAgg) | 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. |