Empir Software Eng (2009) 14:685–719
DOI 10.1007/s10664-009-9106-z

Assessing the understandability of UML statechart diagrams with composite states—A family of empirical studies

José A. Cruz-Lemus · Marcela Genero · M. Esperanza Manso · Sandro Morasca · Mario Piattini

Published online: 17 February 2009

Editor: Dag Sjoberg

Abstract The main goal of this work is to present a family of empirical studies that we have carried out to investigate whether the use of composite states may improve the understandability of UML statechart diagrams derived from class diagrams. Our hypotheses derive from conventional wisdom, which says that hierarchical modeling mechanisms are helpful in mastering the complexity of a software system. In our research, we have carried out three empirical studies, consisting of five experiments in total. The studies differed somewhat as regards the size of the UML statechart models, though their size and the complexity of the models were chosen so that they could be analyzed by the subjects within a limited time period. The studies also differed with respect to the type of subjects (students vs. professionals), the familiarity of the subjects with the domains of the diagrams, and other factors. To integrate the results obtained from each of the five experiments, we performed a meta-analysis study which allowed us to take into account the differences between studies and to obtain the overall effect that the use of composite states has on the understandability of UML statechart diagrams throughout all the experiments. The results

J. A. Cruz-Lemus (✉) · M. Genero · M. Piattini
Department of Technologies and Information Systems, University of Castilla-La Mancha,
Paseo de la Universidad, 4, Ciudad Real 13071, Spain
e-mail: JoseAntonio.Cruz@uclm.es

M. Genero
e-mail: Marcela.Genero@uclm.es

M. Piattini
e-mail: Mario.Piattini@uclm.es

M. E. Manso
Departament of Computer Science, University of Valladolid, Campus Miguel Delibes, E.T.I.C.,
Valladolid 47011, Spain
e-mail: manso@infor.uva.es

S. Morasca
Dipartimento di Scienze della Cultura, Politiche e dell'Informazione,
Università degli Studi dell'Insubria, Como, Italy
e-mail: Sandro.Morasca@uninsubria.it

obtained are not completely conclusive. They cast doubts on the usefulness of composite states for a better understanding and memorizing of UML statechart diagrams. Composite states seem only to be helpful for acquiring knowledge from the diagrams. At any rate, it should be noted that these results are affected by the previous experience of the subjects on modeling, as well as by the size and complexity of the UML statechart diagrams we used, so care should be taken when generalizing our results.

Keywords UML statechart diagrams · Understandability · Composite states · Family of experiments

1 Introduction

Modeling is at the core of many disciplines, but it is especially important in engineering because it facilitates the communication and construction of complex systems from smaller parts (Thomas 2004). Models help us understand a complex problem and its potential solutions through abstraction. This is why software systems, which are often among the most complex of all engineering systems, can greatly benefit from using models and modeling techniques (Selic 2003). This idea is now receiving even more emphasis, since the software industry is moving towards Model-Driven Development (MDD) processes (Atkinson and Kühne 2003), in which software is developed at a higher level of abstraction than source code, based on models and model transformations. The MDD paradigm therefore focuses the effort of development on the design of models, rather than on coding. Correspondingly, the focus of software quality assurance is shifting from system implementation towards system modeling.

To be useful and effective, an engineering model must possess the following five key quality characteristics to a sufficient degree (Selic 2003): abstraction, understandability, accuracy, predictiveness and inexpensiveness.

In this paper, we focus on understandability because it is recognized as one of the main factors influencing maintainability[1], and it is well-recognized that a large part of the effort invested in the development of any software product is devoted to maintenance (Pigoski 1997). More specifically, we focus on the understandability of UML statechart diagrams, since UML has become the *de facto* standard for modeling software systems; added to this is the fact that UML statechart diagrams have become an important technique for describing the dynamic aspects of a software system (Denger and Ciolkowski 2003). UML statechart diagrams are also considered to be one of the most important UML diagrams and they should be used by practitioners as a starting point for training newcomers to UML (Bolloju and Leung 2006).

The main goal of this line of research, which we have pursued over the last 5 years, was to investigate which constructs influenced the understandability of UML statechart diagrams, since a UML statechart diagram must be understood before any desired change on it can be identified, designed, or implemented. In the quest to reach this objective, we carried out a controlled experiment and a replication of it (Cruz-Lemus et al. 2005b). We found that activities, guards, simple states and transitions were the UML constructs that most influenced the understandability of UML statechart diagrams, but that the effect of

[1] Even though understandability is not a maintainability sub-characteristic by the ISO 9126 standard (ISO/IEC, 2001), software quality research considers understandability to be a main factor influencing maintainability (Briand et al. 2001; Fenton and Pfleeger 1997; Harrison et al. 2000).

composites states was not clear. Considering these results as preliminary, we decided to continue investigating composite states.

Composite states allow modelers to structure UML statecharts in a hierarchical fashion. A composite state represents the abstraction of an entire UML statechart diagram into which the composite state can be refined. As such, composite states are an important construct of the UML statechart diagrams metamodel (OMG 2003) and they are believed to be a fundamental modeling abstraction mechanism to help modelers master the complexity of a software system. From a theoretical point of view, UML statechart diagrams with composite states extend finite state machines to facilitate the description of highly complex behaviors (Hu and Shatz 2006) by dividing the system into smaller, less complex parts thereby making this system easier to understand. This in turn leads to a model that is easier to develop and modify.

Taking as a starting point the common use of hierarchical structures in modeling techniques, we thus hypothesized that abstracting a UML statechart diagram composed of highly related simple states and transitions into a composite state could help improve the understandability of a UML statechart diagram. Empirical support needs to be provided to show if this belief is actually true and, if so, under what conditions.

As related works show (see Section 2), references on empirical studies related to dynamic modeling in general and UML statechart diagrams in particular are few and far between. To our knowledge, the influence of composite states on the understandability of UML statechart diagrams has not been studied in the literature previously, despite the importance of the topic. This fact motivated us to gather empirical evidence for our hypothesis.

In this work, we present a family of three empirical studies consisting of five controlled experiments, whose design and execution were gradually modified and improved to alleviate some threats to the validity of the different component studies. We used relatively small statechart diagrams (10 to 25 states) as experimental materials and the experimental subjects were undergraduate and graduate students of Computer Science at several universities, along with a number of professionals with an average of 2 years' experience in UML modeling.

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 recognized 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 (Lipsey and Wilson 2001; Miller 2000; Pickard 2004). 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 to extract conclusions of a more general kind from families of experiments, with or without significant results.

Since we have not evaluated industrial systems with a large range of different size and complexity, we cannot generalize our findings to every usage of composite states in UML statechart diagrams. Nevertheless, our common family of experiments seems to indicate that the use of composite states is not always beneficial.

The paper is organized as follows. Section 2 presents related work. Section 3 provides a roadmap of the family of experiments that we have performed. Section 4 introduces the Cognitive Theory of Multimedia Learning (CTML) (Mayer 2001), which we have used as a background in some of our experiments. Sections 5, 6, 7 then explain in detail the experimental process used to carry out each of the studies that are part of the family of experiments. Section 8 summarizes the threats to the validity of the family of empirical studies. In Section 9 the results of the meta-analysis performed with the data are presented. The main conclusions achieved from this family of experiments and the future work that is planned are in Section 10.

2 Related Work

In this section, we situate our empirical study in relation to some other work found in the relevant literature.

Comprehension has been widely studied. In the literature, we can find works that have studied the comprehension of programs (Woodfield et al. 1981), complete models (Agarwal et al. 1999) or specific diagrams such as UML class diagrams (Purchase et al. 2001, 2002; Yusuf et al. 2007), UML collaboration diagrams (Glezer et al. 2005; Purchase et al. 2001, 2002) and UML sequence diagrams (Glezer et al. 2005; Xie et al. 2007). We can even find examples of pieces of work which study how the use of different artifacts, e.g. stereotypes, affects the way in which models are understood (Genero et al. 2008; Ricca et al. 2007; Staron et al. 2006).

As we have commented previously, understandability is considered to be a main factor influencing maintainability (Briand et al. 2001; Fenton and Pfleeger 1997; Harrison et al. 2000) and we can also find other works taking up this issue (Arisholm and Sjøberg 2004; Genero et al. 2007).

In some of these studies we have found that experience is a factor to be taken into account when measuring comprehension (Arisholm and Sjøberg 2004; Bolloju and Leung 2006; Ricca et al. 2007; Yusuf et al. 2007).

We found the following two papers dealing with empirical studies on the comprehension of UML diagrams which model dynamic aspects of an OO system:

- Otero and Dolado (2004) evaluate the comprehension of the dynamic modeling in UML designs by using two experiments in which they compare the comprehension of UML sequence, collaboration, and statechart diagrams. They conclude that sequence diagrams are the most appropriate for comprehension of management information applications, collaboration diagrams are those best suited to real-time non-reactive systems, and statechart diagrams are the most appropriate for real-time reactive systems.
- Otero and Dolado (2005) present two controlled experiments for evaluating the semantic comprehension of two standard languages, UML versus OPEN Modeling Language (Firesmith et al. 1998), from the perspective of dynamic modeling. The results reveal that the specification of dynamic behavior using OPEN Modeling Language is faster to comprehend and easier to interpret than when using the UML language, regardless of the type of dynamic diagram.

As we commented in the introduction, the main goal of our line of research over the last 5 years has been to investigate which constructs influenced the understandability of UML statechart diagrams, so the most closely related work is that done by ourselves prior to this. We had carried out a controlled experiment and a replication of it (Cruz-Lemus et al. 2005b) in which we found that some of the UML statechart diagram constructs (activities, guards, simple states and transitions) were the ones that most influenced the understandability of UML statechart diagrams. To perform that experiment, a group of teachers and students from the University of Castilla-La Mancha (Spain) performed a series of comprehension tasks on 20 different UML statechart diagrams which covered a broad range of values for the proposed metrics. In this study, composite states did not seem to affect the understandability of UML statechart diagrams.

In addition, in Cruz-Lemus et al. (2005c) we presented an experiment and its replication whose purpose was to find out the optimal nesting level of composite states within UML statechart diagrams. 38 Computer Science students from the University of Murcia (Spain) answered a set of comprehension questions related to the same system, but modeled using 0, 1, and 2 nesting levels in composite states, i.e., without composite states, with one composite state and with composite states within composite states. They concluded that a flat nesting level makes the diagrams more easily understandable.

This review of the literature reveals that the use of composite states and their impact on the comprehension of UML statechart diagrams have not been investigated in depth, despite the need for empirical studying of UML diagram comprehension, and in spite of how many recently- published works there are.

Even though in (Cruz-Lemus et al. 2005b) we found that composite states seem not to affect the comprehension of UML statechart diagrams, we considered this a bit suspicious, so we decided to investigate this finding in greater depth. Starting from the common use of hierarchical structures in modeling techniques, we decided to hypothesize that abstracting a UML statechart diagram composed of highly related simple states and transitions into a composite state could help improve the understandability of a UML statechart diagram. This hypothesis was what led us to carry out the research that we are presenting in the current study.

3 The Family of Experiments

An experiment may be a part of a common family of studies, rather than being an isolated event (Basili et al. 1999). Common families of experiments allow researchers to answer questions that are beyond the scope of individual experiments and let them generalize findings across studies, thus providing evidence for confirming or rejecting specific hypotheses. In addition, common families of studies can contribute to devising important and relevant hypotheses that may not be suggested by individual experiments. A common family of experiments is not necessarily composed only of identical replications of the same study. Materials, hypotheses, and specific tasks assigned to the subjects may be refined across experiments, based on the knowledge obtained after each experiment.

Figure 1 shows the chronology of the family of experiments we have carried out in our study on the understandability of UML statechart diagrams.

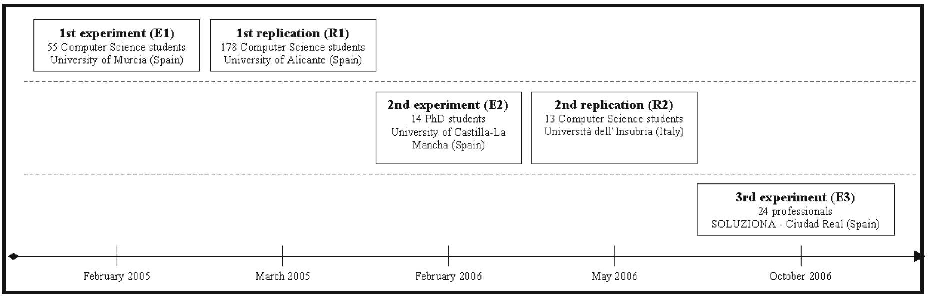

1st experiment (E1)
55 Computer Science students
University of Murcia (Spain)
1st replication (R1)
178 Computer Science students
University of Alicante (Spain)
2nd experiment (E2)
14 PhD students
University of Castilla-La Mancha (Spain)
2nd replication (R2)
13 Computer Science students
Università dell' Insubria (Italy)
3rd experiment (E3)
24 professionals
SOLUZIONA - Ciudad Real (Spain)
February 2005
March 2005
February 2006
May 2006
October 2006

Fig. 1 Chronology of the family of experiments

The first experiment and its replication (E1 and R1) took place in two universities in Spain in 2005. The materials and tasks to be performed were quite simple and the background knowledge of the undergraduate students used as subjects was not advanced. These studies provided some initial results that were later strengthened with the other experiments of the family.

The second experiment and its replication (E2 and R2) took place in two universities, one in Spain and the other in Italy, in 2006. The Italian students' background was similar to that of those in the previous study (E1 and R1), but the Spanish subjects were PhD students and had more experience in modeling. In addition, the materials and tasks assigned to the subjects were improved, especially with the use of the CTML (Mayer 2001) for assessing the complete set of variables of the experimental design. We describe this theory in more detail in Section 4.

In these studies, we used students as experimental subjects. The tasks to be performed did not require high levels of industrial experience, so we believed that this experiment could be considered appropriate, as suggested in the literature (Basili et al. 1999; Höst et al. 2000). Working with students also implies a set of advantages, such as the fact that the prior knowledge of the students is rather homogeneous, there is the possible availability of a large number of subjects (Verelst 2004), and there exists the chance to test experimental design and initial hypotheses (Sjoberg et al. 2005). An additional advantage of using novices as subjects in experiments on understandability is that the cognitive complexity of the objects under study is not hidden by the experience of the subjects.

The main difference between the first four studies (E1, R1, E2, and R2) and the third experiment (E3) lies in the fact that we had professionals as experimental subjects in E3. Another feature that made that experiment distinct was that the materials and tasks were further renewed and improved.

In studies E1 and R1, we used variable *understandability effectiveness*, defined as the ability to understand the presented material correctly. In studies E2, R2 and E3, we added two new variables related to the CTML, *retention* and *transfer*. We explain these variables in Section 4.

These three variables were measured by using three separate tests based on questionnaires. The values of *understandability effectiveness (UEffec), transfer (UTrans),* and *retention (UReten)* were computed as the number of correct answers for each specific test divided by the number of questions.

The time needed to complete a test was also measured, but we chose not to use it because, from our own experience and following the advice of several experts, we have concluded that time is not a good indicator of understandability on its own. It provides information only about how quickly the tasks have been performed, but not about how well.

As for the design of the experiments, we used the guidelines provided in several works (Juristo and Moreno 2001; Kitchenham et al. 2002; Wohlin et al. 2000). Taking into account the kind of experimental designs used and the treatment of the studies, an appropriate statistical method for obtaining the results is an ANOVA (Kirk 1995; Winer et al. 1991). We set a statistical significance threshold $\alpha=0.05$ in all of our studies, so we rejected the null hypotheses of our studies if the statistical tests we used provide a statistical significance (p-value) of the results that was not higher than 0.05. We also studied the power of the statistical test when non-statistically significant results were obtained. We used SPSS (SPSS 2003) to perform all the statistical analyses.

We examine all the threats to validity of the experiments in Section 8.

4 The Cognitive Theory of Multimedia Learning (CTML)

Models in general and conceptual models in particular include both graphics and text. (Mayer 2001) proposed a definition of “multimedia” to include descriptions that include “words” and “pictures”. Conceptual models can be considered multimedia messages, since they include both words and graphic elements (Gemino and Wand 2005).

We have used CTML (Mayer 2001) to explain how individuals viewing explanative material develop an understanding of multimedia content being presented to them. One of the main strengths of this theory lies in the experimental studies that have been based on it to compare text-only presentations with graphics/text presentations in several fields (Craig et al. 2002; Gemino and Wand 2003; Mayer 1989; Mayer and Anderson 1991; Mayer 2001; Tabbers 2004).

There are a number of reasons for choosing CTML as a means of measuring how subjects understand the materials that are being presented (Gemino and Wand 2005). Firstly, CTML focuses on words and graphics, which are the elements used by UML. Secondly, CTML provides principles for the design of effective multimedia presentations that can be empirically tested. In third place, CTML has evolved over a decade of work, in which experimental instruments and methods have been developed (Mayer 1989, 2001).

CTML suggests that a learner is not an “empty vessel” waiting to be filled with domain information, but an active processor with limited cognitive capacity who attempts to integrate presented material with previous knowledge. This implies that individuals might differ in how they understand the same model, depending on prior knowledge and the attention they give to various parts of the model.

(Mayer 2001) suggests that three outcomes are possible when presenting explanative material: (1) no learning, (2) fragmented learning, and (3) meaningful learning. These outcomes are primarily based on concepts that can be measured by two variables that Mayer labels *retention* and *transfer.*

Retention is defined as the comprehension of material being presented. *Transfer* is the ability to use knowledge gained from the material to solve related problems not directly answerable from it. No learning occurs where retention and transfer are low. Fragmented learning occurs where retention is high but transfer is low. This result indicates that material has been received but has not been integrated well with prior knowledge. It suggests that memorization has occurred, rather than meaningful learning. Finally, meaningful learning occurs when both retention and transfer are high. High transfer indicates that information has been integrated into long-term knowledge and a high level of understanding of the presented material has been achieved.

5 First Experiment and Replication (E1 and R1)

In this section, we outline the main characteristics and results of the first experiment (E1) and its replication (R1). More details about this study can be found in (Cruz-Lemus et al. 2005a).

All the subjects received a short training session before the experiment, in which the instructor commented on the main constructs of UML statechart and showed two examples of the experimental tasks to be performed. These examples, as well as those performed in the rest of experiments and replications, were neutral with regards to the independent variable (whether using composite states or not), as one example contained composite states and the other did not.

We split the subjects randomly into two groups, which we here call Group A and Group B. Two different domains were used, one involving the functioning of an ATM (Automated Teller Machine) and the other a phone call. For each domain, two conceptually identical diagrams were used, but while one of the diagrams included composite state(s), the other did not.

In the first part of the experiment, we used the ATM domain, in which the subjects in Group A received a diagram without composite states, while the subjects in Group B received a diagram with composite states. In the second part of the experiment, we used the phone call domain. Subjects in Group A received a diagram with composite states, while the subjects in Group B received a diagram without composite states. The experiment design is summarized in Table 1.

This process of assigning subjects to the 4 different treatments, obtained by combining the dependent variables (Domain and Composite States) corresponds to a 2x2 factorial design with confounded interaction (Winer et al. 1991), because within a domain, the variable Composite States changes together with the group of subjects. So the learning effect is alleviated.

Half of the subjects of each group received the diagram without composite states first and the diagram with composite states second, while the other half received them in inverse order, to avoid possible learning effects.

Each diagram had a test enclosed, with 6 questions. The questions for each domain were the same, regardless of the particular use of composite states. The questions inquired about navigation between states and the effects that it produced.

To increase the motivation and interest on the part of the subjects, the instructor explained to the students that the exercises in the experiment would be similar to those that they would find in their exam at the end of the term. The goal of this experiment and the research question were not disclosed in this experiment (Carter et al. 2003) nor in any of the following studies, however.

In this study, we measured *UEffec*, the *understandability effectiveness* (defined in Section 3) of a set of UML statechart diagrams.

We explain the main differences between experiment E1 and its replication R1 in the following subsections. The main conclusions and threats will be presented.

5.1 First Experiment (E1)

The subjects in this experiment were in the fourth year of Computer Science and had received a complete Software Engineering course in which they had studied modeling techniques, including UML.

The other main features of this experiment are outlined in Table 2.

Table 3 shows the descriptive statistics of the data.

Table 1 E1 and R1 design

		Domain	
		ATM	Phone Call
Composite States	**Without CS**	Group A	Group B
	With CS	Group B	Group A

Table 2 E1 main features

Null-Hypothesis	H0: The use of composite states does not improve the understandability effectiveness of a UML statechart diagram. H1: ¬H0
Location	University of Murcia (Spain)
Date	February 2005
Subjects	55 Computer Science students (28 Group A; 27 Group B)
Dependent Variable	Understandability of UML statechart diagrams, measured by *UEffec.*
Independent Variables	The domain to which the diagrams were related and the use or not of composite states (CS) in it.

Table 3 shows that the subjects obtained better results for *UEffec* when working with those diagrams that did not use composite states. After removing the outlier values found, we also performed an ANOVA test, which is the most appropriate test for exploring the results of a 2x2 factorial design with interaction confounded (Kirk 1995; Winer et al. 1991). The ANOVA results are shown in Table 4.

In Table 4 and the rest of the tables in the document related to ANOVA, we show the results of Fisher's F test, where *Source* column describes the independent variables, *df* refers to the degrees of freedom, *F* is the value of the test statistic, *p-value* is the statistical significance obtained, and *Observed Power* is the estimated power of the test based on α=0.05.

We cannot make any strong conclusion, as at an α level we cannot reject H_0, i.e., there is no effect from the use of composite states. The observed power of the test is low, probably because of a small effect size, so we would be assuming a 0.756 (or 1−0.244) estimated probability of Type II error in our assertions. Even though the results are not conclusive, they seem to indicate that there is no appreciable impact of the use of composite states on the understandability effectiveness of UML statechart diagrams.

5.2 First Experiment Replication (R1)

The subjects in this experiment were in the second year of Computer Science and were not very familiar with modeling or with UML. They were taking their first course of Software Engineering at the time of the experiment.

The main differences with respect to E1 are detailed in Table 5.

Due to limitations of physical space in the classrooms where R1 took place, the subjects were divided into two groups of 92 and 86 subjects respectively and they performed the experiment at different times. To be specific, the second group began and finished 1 h later than the first. Nevertheless, there was no interaction between the subjects of the different groups.

The skills of the subjects using UML for modeling, especially UML statechart diagrams, were much lower in R1 than in E1, as most of them had only a few months of experience,

Table 3 E1 mean and standard deviation (in parentheses) for the *understandability effectiveness*

Domain	ATM		Phone Call	
CS	With (n=27)	Without (n=28)	With (n=28)	Without (n=27)
UEffec	0.9506 (0.0082)	0.9702 (0.0042)	0.9286 (0.0840)	0.9506 (0.1014)

Table 4 E1 ANOVA results

Source	df	F	p-value	Observed Power
Domain	1	1.628	0.205	0.244
CS	1	1.628	0.205	0.244

and they had not worked with some UML metamodel constructs (e.g. composite states) yet. That being the case, the only knowledge they had about composite states was acquired during the introductory session before the experiment.

We used the same techniques and performed the same analysis as in E1. The results obtained are summarized in Table 6.

In this case, most of the subjects that had received diagrams without composite states or the diagram that modeled the ATM with composite states answered all the questions correctly.

Table 6 again shows that the subjects obtained better overall results for the *UEffec* when working with those diagrams that did not use composite states. We removed the outlier values and performed an ANOVA test, whose results are in Table 7.

We can observe that there is not a significant effect from the domain or from the use of composite states and, in this case, the power is still very low. If we rejected the null-hypothesis, we would be assuming a 0.687 estimated probability of Type II error. Once again then, the results are not conclusive although they do seem to indicate that there is no appreciable impact of the use of composite states on the understandability effectiveness of UML statechart diagrams.

5.3 E1 and R1 Conclusions

The main goal of E1 and R1 was to study the effect that the use of composite states had on the understandability effectiveness of UML Statechart Diagrams. Considering the results obtained, we cannot conclude anything definitively, as these were not statistically significant and the values of test power were low. We could notice, nonetheless, that the use of composite states does not seem to significantly improve the understandability effectiveness of UML statechart diagrams.

6 Second Experiment and Replication (E2 and R2)

Given the results of E1 and R1, we reviewed the experimental process that had been carried out in that study, focusing especially on the design and the materials that had been given to the subjects. After reading about the experimental approaches performed in other works (Bodart et al. 2001; Gemino and Wand 2005), we sought to enrich the type of tasks that the subjects had been required to carry out in our empirical studies, to reflect fully the understanding that the subjects had on the diagrams. We hence decided to carry out another

Table 5 R1 main differences with E1

Location	University of Alicante (Spain)
Date	March 2005
Subjects	178 Computer Science students (89 in each group)

Table 6 R1 mean and standard deviation (in parentheses) for the *understandability effectiveness*

Domain	ATM		Phone Call	
CS	With (n=89)	Without (n=89)	With (n=89)	Without (n=89)
UEffec	0.9663 (0.0719)	0.9625 (0.1028)	0.8978 (0.1188)	0.9491 (0.0991)

experiment, after reviewing the design and the materials that would be given to the subjects. As we have commented above, the new approach that we decided to use was based on CTML (Mayer 2001) (See section 4).

We carried out a controlled experiment (E2) and a replication (R2) in which, in addition to *UEffec*, we took into account the two variables presented by CTML: *UTrans* and *UReten.*

In E2 and R2, all subjects received a short training session before the experiment, in which the main constructs of UML statechart diagrams were explained. Several of the subjects had not used UML statechart diagrams for a while. Some examples, similar to the tasks to be performed in the experiment, were also explained by the instructor of the experiment, so that the subjects had a clear idea of how to do the experimental tasks.

As in E1 and R1, we used four different diagrams. They modeled two different domains (an ATM and an alarm clock). We chose these two domains because it was our opinion that there should be a non-negligible difference in the degree of familiarity of the subjects with each domain. In particular, we believed that the alarm clock domain was more complex than the ATM, at least for the dynamic behavior modeled in the diagrams. One possible problem with E1 and R1 may have been that the diagrams were quite easy, so the use of composite states would not actually make any appreciable difference.

In E2 and R2, the design and type of statistical study is identical as in E1 and R1 (Section 5). For each domain, we used two different diagrams with an identical semantic content, one with composite states and the other without. Each subject received two diagrams, one with and another without composite states. Each of them related to a different domain. Thus, we obtained two different groups, as shown in Table 8.

Each subject had to perform three questionnaire-based tests, each about a different variable we studied:

- Test 1 contained 7 questions which were exactly the same within each domain, independent of the usage of composite states. The questions inquired about navigation between states, variable values, etc. The subjects were allowed to check the diagram to answer the question. This is a kind of task we had already used in previous studies (Cruz-Lemus et al. 2005a; Cruz-Lemus et al. 2005b). With this test, we studied the *UEffec* variable.
- Test 2 consists of a questionnaire with 5 questions in which the subjects were asked about how the model worked, i.e., some questions that were more specific than in the previous test. In this case, the subjects were not allowed to look at the diagrams to answer the questions, as these had been removed previously. This task allowed us to measure the *UTrans* variable.

Table 7 R1 ANOVA results

Source	df	F	p-value.	Observed Power
Domain	1	1.873	0.183	0.260
CS	1	2.340	0.139	0.313

Table 8 E2 and R2 group design

		Domain	
		ATM	Alarm Clock
CompositeStates	Without CS	Group A	Group B
	With CS	Group B	Group A

- Test 3 consists of a 'fill-in-the-blanks' task. The subjects received a text in which the requirements of the model were commented on, but there were a number of missing words. The subjects had to fill in these blanks without using the diagrams, which had not been given back to them. With this task we studied the *UReten* variable.

These two types of tests were similar to some others used in similar studies (Gemino and Wand 2005; Khatri et al. 2006) which deal with model comprehension using the CTML.

E2 and R2 started with a 25 min introductory session in which the instructor explained the main constructs of a UML statechart diagram. We then showed two examples in a shortened version, along with the correct answer to each question.

Throughout this time, the subjects were allowed to ask the instructor about any doubts they might have; they could also make any remarks they wished to.

We randomly assigned the subjects to two groups. Then, each subject received a diagram, depending on particular subject group he /she belonged to, along with the corresponding sheet for Test 1. From that moment on, the subjects had 20 min to look at the diagram, try to understand how the model worked, and answer the questions.

When that task was completed, these materials were collected and each subject received the sheets with Tests 2 and 3 for the diagrams that they had been studying. They had 20 min to work on both tests.

The materials were collected once more and the above process was repeated, i.e., first they received a diagram and a Test 1 sheet and then later tests 2 and 3. In this second diagram, each subject received a different domain than in the first one (ATM / clock) and also a different usage of composite states (with / without).

The process is represented in diagram form in Fig. 2.

In the following subsections, we present the specific details for E2 and R2.

6.1 Second Experiment (E2)

The main features of E2 can be found in Table 9.

After collecting the data, we first carried out an analysis of the descriptive statistics. In Table 10, we present the means and the standard deviations across the different groups.

The values for *UEffec* are lower than in the previous studies (E1 and R1). A possible explanation for this is that, in this case, the materials had been modified and there were more questions. Besides that, the questions were more difficult to answer.

There is a clear trend for the *UEffec* variable: the subjects obtained better results in those diagrams modeled without using composite states, regardless of the domain of the diagram. Nevertheless, for the *UTrans* variable we find the opposite situation; here, better results are obtained for those diagrams modeled using composite states. In the third case, for the *UReten* variable, the results were different, depending on the domain.

We detected outlier values in the different tests and decided to exclude them from the data analysis. We then proceeded to test the previously described statistical hypotheses

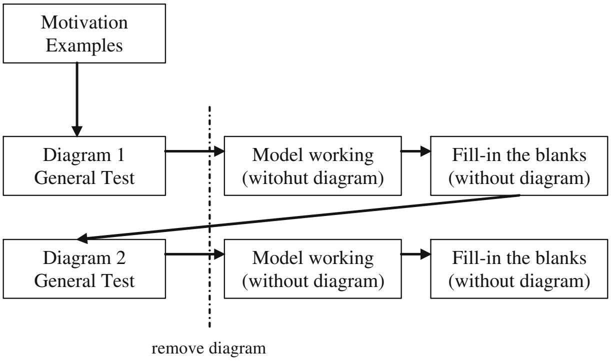

Motivation Examples
Diagram 1 General Test
Model working (witohut diagram)
Fill-in the blanks (without diagram)
Diagram 2 General Test
Model working (without diagram)
Fill-in the blanks (without diagram)
remove diagram

Fig. 2 E2 and R2 procedure overview

through an ANOVA test, which is the most appropriate test for exploring the results of a 2x2 factorial design with interaction confounded (Kirk 1995; Winer et al. 1991). The results are shown in Table 11.

In all cases (*UEffec*, *UTrans* and *UReten*), the variables are not significantly affected by the domain or the use of composite states. The test powers are low, so the possibility of producing an error by accepting the null-hypotheses is high. The results are therefore not conclusive.

In this case, the results obtained for the three variables agree with those obtained in E1 and R1 for *UEffec*.

6.2 Second Experiment Replication (R2)

The main differences with respect to E2 are detailed in Table 12.

Table 13 presents the means and the standard deviations across the different groups used in R2.

In this study, the only variable that shows a trend in its results, independent of the domain of the diagrams, is the *UTrans* variable. The results obtained by the subjects were better for those diagrams modeled without using composite states. In the other two cases, the subjects obtained different results, depending on the domain and the use or not of composite states.

Table 9 E2 main features

Null-Hypotheses	H_{0a}: using composite states does not improve *UEffec* in subjects when they are trying to understand a UML statechart diagram. H_{1a}: $\neg H_{0a}$ H_{0b}: using composite states does not improve *UTrans* in subjects when they are trying to understand a UML statechart diagram. H_{1b}: $\neg$ H_{0b} H_{0c}: using composite states does not improve *UReten* in subjects when they are trying to understand a UML statechart diagram. H_{1c}: $\neg H_{0c}$
Location	University of Castilla-La Mancha. Ciudad Real (Spain)
Date	February 2006
Subjects	14 PhD students (7 in each group)
Dependent Variable	Understandability of UML statechart diagrams, measured through the measures of *understandability effectiveness, retention* and *transfer.*
Independent Variables	The domain of the UML statechart diagrams and the use or not of composite states (CS) when modeling the diagram.

Table 10 E2 Means and standard deviations (in parentheses) across groups

Domain	ATM		Alarm clock	
CS	With (*n*=7)	Without (*n*=7)	With (*n*=7)	Without (*n*=7)
UEffec	0.6837 (0.0761)	0.7959 (0.0687)	0.7857 (0.0643)	0.8673 (0.0453)
UTrans	0.6429 (0.0684)	0.6349 (0.0898)	0.6905 (0.0765)	0.5774 (0.0683)
UReten	0.6071 (0.0798)	0.6143 (0.0643)	0.5714 (0.1359)	0.4444(0.1118)

Comparing these results with those obtained in E2, we can observe an increase in the means of the variables. A possible reason for that is the small sample size in both studies, together with the fact that they are not randomized. These facts probably indicate that we could find two groups with different backgrounds in our population. After removing the outliers, we also performed an ANOVA test with the data obtained in R2. Table 14 summarizes the results obtained.

Again, in all cases (*UEffec, UTrans* and *UReten*), the results indicate that they are not significantly affected by the domain or the use of composite states. Once more, the test powers are low, so the possibility of producing an error by accepting the null-hypotheses is high. The results are not conclusive, therefore.

6.3 E2 and R2 Conclusions

The main goal of E2 and R2 was, once more, to study the effect that the use of composite states had on the understandability of UML Statechart Diagrams. Considering the results obtained in E1 and R1, we revised the materials and introduced new variables, included in the CTML.

After reviewing the results obtained, we again cannot come to any definitive conclusions, as they were not statistically significant and the values of test power were low. It is true, nonetheless, that in this case, we could notice that the use of composite states does not seem to significantly improve the understandability transfer of UML statechart diagrams.

7 Third Experiment (E3)

In this section, we explain the process we followed when we carried out the third member of the family of experiments.

In this experiment, we further revised and improved the materials and tasks to perform. More importantly, software professionals were involved as subjects in this experiment.

Table 11 E2 ANOVA results

		UEffec			UTrans			UReten		
Source	df	F	p-value	Observed Power	F	p-value	Observed Power	F	p-value	Observed Power
Domain	1	1.873	0.183	0.260	0.004	0.948	0.050	1.043	0.317	0.166
CS	1	2.340	0.139	0.313	0.643	0.430	0.120	0.354	0.557	0.088

Table 12 R2 main features

Location	Università degli Studi dell'Insubria, Como (Italy)
Date	May 2006
Subjects	13 undergraduate Computer Science students (6 or 7 in each group)

The experiment was carried out in the facilities of the Soluziona Software Factory Company, located in Ciudad Real, Spain. Soluziona, which now belongs to the INDRA Corporation, currently holds a top position in the market of professional services in software, with a sales volume which is close to 800 million Euros and, after a long period of expansion, the company has spread to over 28 countries in 4 different continents. The company has recently reached maturity level 3, according to the CMMi model, and it is planned for level 4 to be achieved in 2009.

7.1 E3 Design

Table 15 outlines the main features of E3.

In this study, there was only one domain, a digital watch (Webb 2006), which has a size and complexity that are representative of a real-life case. In this case, we have used a randomized blocks design to control the effect of the subjects' experience over the variable CS.

The working hypotheses and part of the procedure to follow were similar to E2 and R2. In the following sections, we explain in more detail the experimental procedure and the results we obtained.

7.2 E3 Procedure

The experiment was divided into two sessions, over 2 days. The first session took place on the afternoon of the first day and the second session the following morning.

In order for the subjects to have a knowledge background that was homogenous, the first session began with a seminar about "Dynamic Modeling with UML." Twenty-five professionals attended the first session and they were provided with a summary of the main concepts of dynamic aspects in modeling in general and in UML in particular. The last part of the seminar focused on UML statechart diagrams, although there was no explicit mention made of any aspect to make the subjects guess the relationship between the seminar and the ensuing experiment.

After the seminar, the instructor explained several UML statecharts with questionnaires, as examples of the test that the subjects were going to perform in that session (Test 0).

Table 13 R2 Means and standard deviations (in parentheses) across groups

Domain	ATM		Alarm clock	
CS	With (n=7)	Without (n=6)	With (n=6)	Without (n=7)
UEffec	0.9082(0.0895)	0.9167 (0.1311)	0.9476 (0.1075)	0.8673 (0.01197)
UTrans	0.5556 (0.2128)	0.7407 (0.0907)	0.6667 (0.1900)	0.7262 (0.2136)
UReten	0.5571 (0.1272)	0.6167 (0.1329)	0.6296 (0.2781)	0.5873 (0.2775)

Table 14 R2 t-test results

		UEffec			UTrans			UReten		
Source	df	F	p-value	Observed Power	F	p-value	Observed Power	F	p-value	Observed Power
Domain	1	0.367	0.550	0.089	0.437	0.515	0.097	0.066	0.800	0.057
CS	1	0.110	0.743	0.062	2.807	0.107	0.362	0.010	0.919	0.051

These examples consisted of questions about navigation through several statechart diagrams. After this, the subjects performed Test 0. This test was used to put the subjects into balanced groups, depending on their knowledge and performance.

They also filled in an anonymous subjective questionnaire in which they included some personal data (age, gender…) and their experience in modeling, OO programming, use of UML, etc. These data indicated that although most of them had developed OO software, only half of the subjects had previously used UML in real projects, and this only once or twice. The average length of experience in OO development was 2 years.

This first session lasted approximately 2 h. After this, all of the Test 0 questionnaires were analyzed and the subjects were assigned to two groups, depending on their results.

The subjects were ordered according to the number of correct answers and the time spent on the questionnaire, as suggested in (Otero and Dolado 2004). After that, those subjects who occupied an odd rank were assigned to Group A, and the others to Group B. Thus, we obtained two balanced groups, as Table 16 shows.

The second session took place the following morning. One of the subjects who had been assigned to Group A did not show up for the second session. We decided not to rearrange the groups, which would now have the same number of subjects. The subjects were informed that they had been grouped depending on their performance in the test collected the previous day.

The second session was composed of three tests, as in experiments E2 and R2. First, the subjects received one UML statechart diagram and a copy of Test 1. To avoid possible learning effects, we adopted a balanced between subjects and blocked design, i.e. each subject was assigned only one diagram. The subjects of Group A received a diagram that was modeled using composite states and those in Group B received exactly the same

Table 15 E3 main features

Null-Hypotheses	H_{0a}: using composite states does not improve *UEffec* in subjects when they are trying to understand a UML statechart diagram. H_{1a}: ¬ H_{0a} H_{0b}: using composite states does not improve *UTrans* in subjects when they are trying to understand a UML statechart diagram. H_{1b}: ¬ H_{0b} H_{0c}: using composite states does not improve *UReten* in subjects when they are trying to understand a UML statechart diagram. H_{1c}: ¬ H_{0c}
Location	Ciudad Real (Spain)
Date	October 2006
Subjects	24 professionals (12 in each group)
Dependent Variable	Understandability of UML statechart diagrams, measured by *UEffec, UTrans,* and *UReten*.
Independent Variables	The use or not of composite states (CS) in the diagrams.

system but this was modeled without using composite states. The 10 questions in Test 1 for both groups were exactly the same.

As in E2 and R2, this test was used to measure the *UEffec* of the model. The questions in Test 1 covered all the different parts of the diagram so that we could make sure that all parts of the diagram had been covered by the subjects before we removed them. This phase lasted for 25 minutes.

When that phase was completed, all the diagrams and tests were collected and Tests 2 and 3 were handed out. Test 2 was used to measure the *UReten* variable and consisted of a fill-in-the-blanks text with 10 gaps that the subjects had to complete in order to build the text with the specifications of the system. The subjects had 15 min for this phase.

Test 3 was used to measure the *UTrans* variable and consisted of a list of 6 tasks to perform, based on the information taken from the diagram. As these tasks were the most complicated part of the experiment, the subjects had 35 min to solve them.

At the end of Test 3, all the materials were collected and the subjects were handed out a debriefing questionnaire, to collect their impressions about the difficulty of the tests and the main positive and negative points that they had found during the experiment.

Figure 3 summarizes and describes all this process in diagram form.

Appendix A contains the diagrams and tests 1, 2 and 3 of this experiment as an example. The diagrams and tests used during the experiment were in Spanish. We have translated them into English here for the reader's convenience.

7.3 E3 Data Analysis and Interpretation

As in the previous experiments, we carried out an analysis of the descriptive statistics of the data. Table 17 presents the means and the standard deviations for the measures of the dependent variables studied in E3.

We can observe how, in this case, the results obtained for the *understandability effectiveness* and *transfer* variables are higher when the subjects worked with the diagram modeled with composite states, while the *retention* variable was higher in the diagram modeled without using composite states.

After removing the outlier values, we performed an ANOVA to test the two unmodified original hypotheses shown in Table 15 and the new H'_{0c} null-hypothesis as we did with the previous analyses. When two groups are compared, an ANOVA produces the same results as a t-test (Kirk 1995; Winer et al. 1991), which is the most common statistical test used for analyzing two groups and one factor. The results obtained are set out in detail in Table 18.

The results shown in Table 18 indicate that there is a statistically significant effect of the use of composite states on the understandability of UML statechart diagrams, in all three variables used (*UEffec, UTrans* and *UReten*).

Using composite states improves the *UEffec* and *UTrans* of the diagram, and worsens the *UReten*. This means that composite states are useful for a better comprehension of the

Table 16 Means and standard deviations for groups after Test 0

Group	N	Correct Answers		Time	
		Mean	Std. Dev.	Mean	Std. Dev.
A	13	5.3846	1.3409	628.62	122.88
B	12	5.4583	1.0967	620.67	135.18

Fig. 3 E3 procedure schema

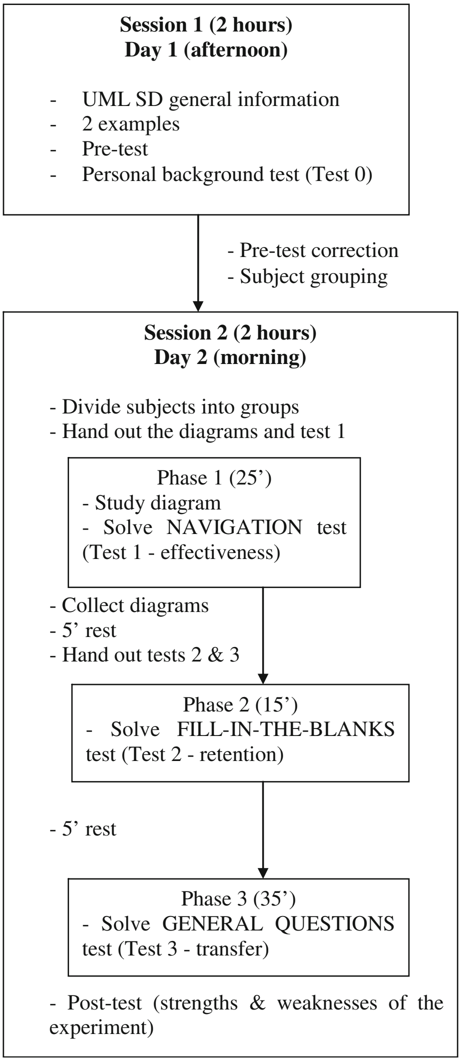

Session 1 (2 hours)
Day 1 (afternoon)
- UML SD general information
- 2 examples
- Pre-test
- Personal background test (Test 0)
- Pre-test correction
- Subject grouping
Session 2 (2 hours)
Day 2 (morning)
- Divide subjects into groups
- Hand out the diagrams and test 1
Phase 1 (25')
- Study diagram
- Solve NAVIGATION test (Test 1 - effectiveness)
- Collect diagrams
- 5' rest
- Hand out tests 2 & 3
Phase 2 (15')
- Solve FILL-IN-THE-BLANKS test (Test 2 - retention)
- 5' rest
Phase 3 (35')
- Solve GENERAL QUESTIONS test (Test 3 - transfer)
- Post-test (strengths & weaknesses of the experiment)

diagram (*UEffec*) and for performing tasks related to the diagram, but not directly answerable from it (*UTrans*). At the same time, they are not useful for memorizing the diagrams (*UReten*).

These results contrast with those obtained in the experiments presented previously. In this case we again intended to assess how composite states affected the understandability of

Table 17 E3 Means and standard deviations (in parentheses)

CS	With (n=12)	Without (n=12)
UEffec	0.7500 (0.1446)	0.6417 (0.1564)
UTrans	0.3690 (0.2079)	0.2619 (0.1677)
UReten	0.7750 (0.1390)	0.8917 (0.0515)

Table 18 E3 ANOVA results

		UEffec			UTrans			UReten		
Source	df	F	p-value	Observed Power	F	p-value	Observed Power	F	p-value	Observed Power
CS	1	257.628	**0.000**	1.000	34.442	**0.000**	1.000	762.338	**0.000**	1.000

UML statechart diagrams, but in this case we used real practitioners instead of students. The complexity of the tasks to be performed was also increased. Our conviction is that these two factors must have affected the results obtained. As well as all these factors, it should be remembered that we used the skills of the subjects to balance their distribution into groups.

8 Threats to the Validity of the Family of Empirical Studies

In this section, we explain some issues that can threaten the validity of experiments, considering the four types of threats proposed in Wohlin et al. (2000).

8.1 Conclusion validity

In E1, R1, E2 and R2, the statistical power was low. As we have already commented, this fact does not allow us to reject erroneous hypotheses without a large degree of uncertainty.

8.2 Internal validity

The number of subjects involved was not large. However, a clear trend was identifiable in only one case.

Our final consideration is that composite states seem to be a construct that requires a certain maturity level to be used properly. As we commented in section 2, experience is a factor to be taken into account when measuring comprehension (Arisholm and Sjøberg 2004; Bolloju and Leung 2006; Ricca et al. 2007; Yusuf et al. 2007). We believe that students probably have not acquired this maturity yet, while practitioners have. This may have been a determining factor in obtaining the results presented here.

8.3 Construct validity

Our measures were built on the basis of the guidelines provided in CTML, and we believe that in this way we have measured the variables appropriately.

8.4 External validity

The diagrams that were used in this study represent relatively simple models and it is certainly possible that if industrial-strength diagrams had been used, different results might have been obtained.

Our results may be applied to UML statechart diagrams and subjects with similar characteristics to those we have presented. These results may be generalized to the entire population of designers who use UML statechart diagrams only after further studies confirm them.

9 Meta-Analysis Study

In Sections 5, 6, 7 of this work, we have presented 5 experiments to investigate the influence of composite states on the understandability of UML statechart diagrams. Table 19 summarizes the ANOVA results studying the effects of the domain and the use of Composite States (CS) on the three dependent variables.

Only in experiment E3 did we obtain statistically significant results that showed that use of composite states improves the way that subjects directly understand how the diagram works (*UEffec*), as well as the performance of tasks related to the diagram, acquiring knowledge from it (*UTrans*). But the use of composite states is not useful for memorizing the diagrams (*UReten*).

As we have seen, no conclusive results were obtained from the individual data analysis, so we decided to integrate them. There are several statistical methods that allow us to accumulate and interpret a set of results obtained through different experiments that are inter-related because they check similar hypotheses (Glass et al. 1981; Hedges and Olkin 1985; Rosenthal 1986; Sutton et al. 2001; Wolf 1986). In the present study, we use meta-analysis because it allows us to extract more general conclusions, even though some of the experimental conditions are not exactly the same.

Meta-analysis is a set of statistical techniques for combining the different effect sizes of the experiments to obtain a global effect of a factor. As measures may come from different environments and not be homogeneous, a standardized measure of each one needs to be obtained and then those measures for estimating the global size effect of the factor must be combined. In our study, the factor is the use of composite states and how that affects UML statechart diagrams understandability.

To carry out the meta-analysis presented in this work we used the Meta-Analysis v2 tool (Biostat 2006). In this meta-analysis we used the mean value for CS(with) minus the mean value for CS(without), and from these values we obtained Hedges' g metric (Hedges and Olkin 1985; Kampenes et al. 2007), which we used as standardized measure. This value expresses the magnitude of the treatment effects, CS, in our case, relative to the within-group standard deviations. It can be used to synthesize studies that have quantified treatment effects in different scales (Rosenthal 1994).

The Hedges' g metric is a weighted mean whose weights depend on sample size (Eq. 1)

$$\overline{Z} = \frac{\sum_i w_i z_i}{\sum_i w_i} \tag{1}$$

where $w_i=1/(n_i-3)$ and n_i is the sample size of the i-th experiment.

The higher the value of Hedges' g is, the higher the corresponding mean difference is too. For studies in Software Engineering, we can classify effect sizes into three different values: small, medium and large (Kampenes et al. 2007).

Once the overall effect size is calculated, we can provide a confidence interval or a p-value which allows us to decide about the meta-analysis hypotheses, such as we can find in other Empirical Software Engineering works (Dybå et al. 2007; Hayes 1999; Laitenberger et al. 1999; Miller and McDonald 1998; Porter and Johnson 1997).

Table 19 ANOVA summary (in bold, significant results)

	UEffec ANOVA				*UTrans* ANOVA				*UReten* ANOVA			
	Domain		CS		Domain		CS		Domain		CS	
	p-value	Observed Power	p-value	Observed Power	p-value	Observed Power	p-value	Observed Power	p-value	Observed Power	p-value	Observed Power
E1	0.205	0.244	0.205	0.244	—	—	—	—	—	—	—	—
R1	0.183	0.260	0.139	0.313	—	—	—	—	—	—	—	—
E2	0.183	0.260	0.139	0.313	0.948	0.050	0.430	0.120	0.317	0.166	0.557	0.088
R2	0.550	0.089	0.743	0.062	0.515	0.097	0.107	0.362	0.800	0.057	0.919	0.051
E3	—	—	**0.000**	1.000	—	—	**0.000**	1.000	—	—	**0.000**	1.000

Our meta-analysis hypotheses can be stated as:

- H_{0a}: using composite states (CS) does not influence the *UEffec*. H_{1a}: $\neg H_{0a}$
- H_{0b}: using composite states (CS) does not influence the *UReten*. H_{1b}: $\neg H_{0b}$
- H_{0c}: using composite states (CS) does not influence the *UTrans*. H_{1c}: $\neg H_{0c}$

Table 20 summarizes the results we obtained with our meta-analysis. For each study and domain within the study, we report the values of Hedges' g and effect size. Specifically, the cells related to effect size contain two pieces of information.

- An indication of the magnitude of effect size, classified as Small, Medium, or Large. The magnitude of effect size is computed based on the standardized difference between two means. For instance, an effect size of 0.5 indicates that the mean of the CS (with) is half a standard deviation larger than the mean of CS (without). Considering the *UEffec* variable, a positive effect means that using composite states improves the understandability effectiveness, whilst a negative effect would mean the opposite. For instance, there is a negative effect size for *UEffec* in E1 with the ATM domain, as denoted by the negative value of Hedges' g, while there is a positive effect size in R1 with the ATM domain, as shown by the positive value of Hedges' g. The same applies to the other two variables. For studies in Software Engineering, we can consider that effect sizes within 1.01 and 3.40 are large; sizes within 0.38 and 1.00 are medium and those within 0 and 0.37 are small (Kampenes et al. 2007).
- An indication of whether the result is statistically significant (S) or not (NS). Note that the global effect is only significant for *UEffec*, in our study.

For the reader's convenience, we show our meta-analysis results in diagram form, as provided by the Meta-Analysis v2 tool (Biostat 2006). Figures 4, 5, 6, one figure for each dependent variable (*UEffec, UTrans, UReten* respectively), display Hedges'g metric with a confidence interval of 95%. Not all the studies contribute equally to the overall conclusion, which is represented by the diamond in the last row of the figures. Each of them receives a specific weight in the meta-analysis, i.e., the study's effect size, represented by the squares in the figures. The estimations for studies with a large sample size are more accurate, so

Table 20 Hedges' g metric values

Study	Domain	*UEffec*		*UTrans*		*UReten*	
		g	Effect size	g	Effect size	g	Effect size
E1	ATM	-2.986	Large S	N/A		N/A	
	Phone call	-0.233	Small NS				
R1	ATM	0.042	Small NS				
	Phone call	-0.467	Medium S				
E2	Alarm clock	-1.345	Large S	1.460	Large S	0.955	Medium NS
	ATM	-1.450	Large S	0.093	Small NS	-0.092	Small NS
R2	Alarm clock	1.023	Large S	-0.272	Small NS	0.142	Small NS
	ATM	-0.072	Small NS	-1.021	Large NS	-0.427	Medium NS
E3	Digital watch	0.695	Medium NS	0.548	Medium NS	-1.075	Large S
Global Effect Size		-0.333	Small S	0.194	Small	-0.193	Small

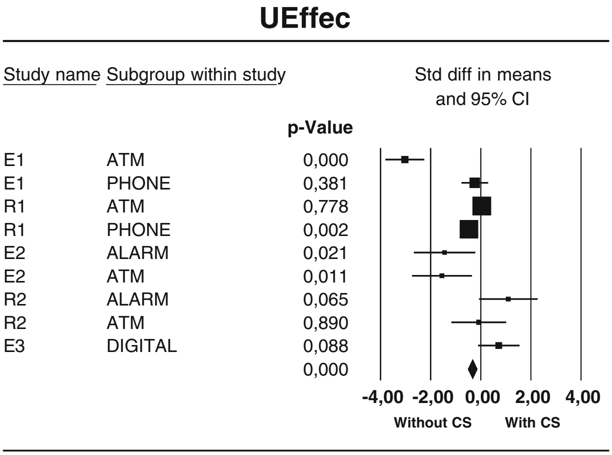


Fig. 4 *UEffec* meta-analysis

they contribute more to the overall effect. However, sample size is not the only factor contributing to the weight of a study. The weight of a study is proportional to the area of the corresponding square in the figures.

After this process, we carried out a new meta-analysis based on two facts. Firstly, the description of E3 and threats to validity make it different from the others, as we used practitioners and more difficult tasks and the design was blocked by using the subjects' experience. Furthermore, the error observed in the three meta-analyses was the highest.

Table 21 summarizes the results including and excluding E3. We can observe how Hedges' g estimation is modified.

Nevertheless, the conclusions, based on the p-values are similar to those obtained previously. The meta-analysis conclusions are thus the following:

- Using composite states makes the understandability effectiveness (*UEffec*) of UML statechart diagrams decrease, with a medium-size effect size (−0.383), i.e., the *UEffec* mean when not using composite states is larger by 0.383 times the standard deviation than when using CS, and the p-value is 0.000. This effect size is lower when we consider all the experiments; because in E3 the composite states improved the

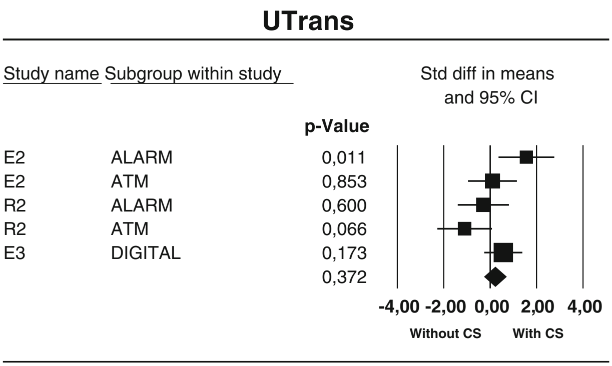


Fig. 5 *UTrans* meta-analysis

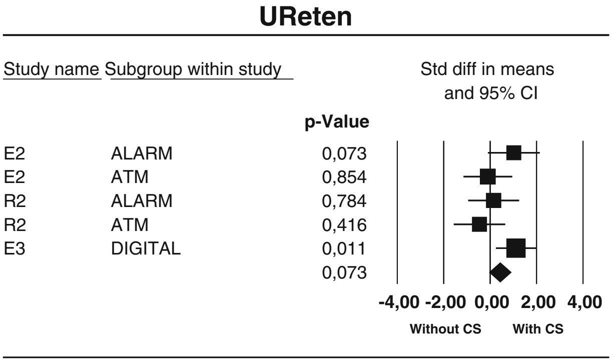


Fig. 6 *UReten* meta-analysis

understandability. We suspect that this positive effect in E3 is due to the use of practitioners as experimental subjects.

- Using composite states has no influence when performing tasks related to the diagram (*UTrans* p-value=0.890). When we include E3, the effect size and the p-value are more in favour of the idea that composite sates help improve the transfer, but these are not enough to be significant.
- Finally, the use of CS has no influence in memorizing the diagrams (*UReten* p-value=0.597). In this case, when including E3 the effect size and the p-value are inclined to indicate that composite states have influence, but this is not significant.

10 Conclusions and Future Work

In this work, we have presented a family of empirical studies to study whether the use of composite states affects the understandability of UML statechart diagrams. We pursued this goal, wishing to obtain empirical evidence on whether the use of composites states is beneficial. This evidence can be used as advice to software engineers or modelers when they are modeling or maintaining object oriented systems using UML.

As suggested in several empirical works related to model comprehension (Bodart et al. 2001; Gemino and Wand 2005) we measured understandability through three measures: Understandability Effectiveness, Retention and Transfer. Each of these measures captures different aspects of the understanding of models when modelers or software engineers deal with models.

Table 21 Meta-analysis results with all the experiments and excluding E3

	UEffec			UTrans			UReten		
	g	Effect size	p-value	g	Effect size	p-value	g	Effect size	p-value
CS excluding E3	-0.383	Medium	0.000	0.037	Small	0.890	0.137	Small	0.597
CS including E3	-0.333	Medium	0.000	0.194	Small	0.384	0.193	Small	0.382

In our empirical work, we have followed the steps suggested in the Empirical Software Engineering field (Tichy 2000), beginning our study with students, to test the original designs, and gradually improving the materials used and the experience of the subjects until performing the last study with practitioners.

The results obtained are valid in the context of relatively simple statechart diagrams (10 to 25 states) and undergraduate students and novice practitioners.

After testing our hypotheses in each individual study, through ANOVA, we could not reach conclusive findings, given that in some cases the results are conflicting across our experiments. We therefore decided to integrate the empirical data through a meta-analysis study.

The main findings obtained through the common family of empirical studies are:

- Our first idea and the one most commonly accepted in the Software Engineering field, was that using composite states helps make UML statecharts more comprehensible. But the meta-analysis results show that using composite states has a negative influence on the understandability effectiveness (*UEffec*) of the diagrams, i.e. the way that subjects directly understand how the diagram works. This finding goes against conventional wisdom. Nevertheless, the particular results of E3 are in favour of this assertion. We suspect that the reason could be that experienced subjects are able to take advantage of the benefits of using composite states. When there is a lack of experience, it might be more difficult to understand and handle the use of composite states. It also seems that the more complicated the tasks to perform are, the more useful the use of composite states is.
- The overall results do not show a clear effect, either in using or in not using composite states related to the concepts of Transfer*,* i.e., the ability to use knowledge gained from the material to solve related problems not directly answerable from it (*UTrans*), and Retention*,* i.e., the ability to memorize the material being presented (*UReten*). But the particular results obtained in E3 show that composite states improve the transfer of the diagrams. As we have just commented, we suspect that the reason could be that when subjects are more experienced, they can take advantage of using composite states, otherwise the effect is the opposite. When looking at retention (*UReten*), the results in experiment E3 present a negative effect and it seems that is better not to use composite states for memorizing diagrams.

Even though the meta-analysis seems to improve the findings for the individual studies, after a rigorous and long period of experimentation we can not provide conclusive findings on whether composites states are beneficial to the understanding of UML statechart diagrams or not, in the context mentioned. However, our results show that the use of composite states may not be always beneficial, as might have been believed after only casual consideration.

At any rate, further investigation is needed in the following directions:

- study of the hypothesis that we have expressed about the effect that composite states may have on subjects without skills in their use;
- extension of the number of practitioners in future studies, which would strengthen the validity of conclusions, as the sample size of E3 (24 practitioners) is small compared to other studies in the family;
- use of more complex diagrams and tasks, from real projects, as we suspect that the use of composite states could be more beneficial when understanding more complex UML statechart diagrams used in real-time systems.

Acknowledgements This research is part of the IDONEO project (PAC08-0160-6141) financed by "Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha" and the ESFINGE project supported by the "Ministerio de Educación y Ciencia (Spain)" (TIN2006-15175-C05-05). The research presented in this paper has been partially funded by the IST project "QualiPSo," sponsored by the EU in the 6th FP (IST-034763), the FIRB project "ARTDECO," sponsored by the Italian Ministry of Education and University, and the project "La qualità nello sviluppo software," sponsored by the Università degli Studi dell'Insubria.

The authors would like to thank:

- Professors Ambrosio Toval (University of Murcia) and Cristina Cachero (University of Alicante) and their students for having cooperated in the performance of the experiments.
- Our PhD students at the Universidad de Castilla-La Mancha (Spain) and our undergraduate students at the Università degli Studi dell'Insubria (Italy) for their unselfish help in the second experiment and its replication.
- The staff in INDRA (formerly Soluziona)—Ciudad Real (Spain) for their time and understanding during the preparation for and performance of the third experiment.
- The reviewers involved in the development process of this paper; all your valuable comments have helped us improve its quality.

Appendix A. Experimental Material

All the experimental material concerning the family of experiments is available on-line at http://alarcos.esi.uclm.es/CSExperiments/

By way of example, we include the diagrams and tests used in the third experiment (E3).

PHASE 1: The Xholon Watch

The Xholon Digital Watch (© 2005, 2006 Ken Webb) sample application simulates the internal structure and behavior of a fairly generic digital watch.

It has four buttons which can be labeled S1, S2, S3 and S4. It displays the time or the date. By pressing the four buttons in various combinations, the current second, minute, hour, month, day and date can be updated, and various other internal functions can be managed. Figure 7

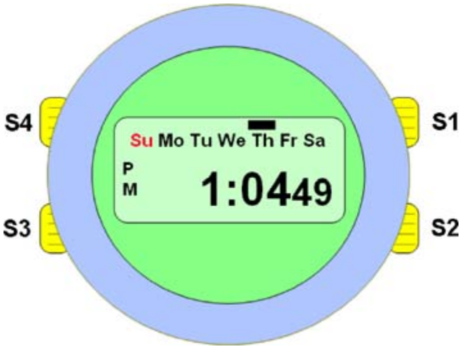


Fig. 7 The Xholon Watch

The main functionalities of the buttons are outlined in the following table:

Button	Functionality
S1	Time / Date
S2	Set Alarm on / off
S3 normal	Chronometer
S3 long	Date/Time/Alarm Update
S4	Light

On the following page, you will find the state machine that modeled how the watch works. Please **look at this** and try to **understand it.** Take as much time as you need.

Once you are ready, start answering the questions in test #1. You must answer the questions **in the same order as they are given** and note down the exact time at which you finish answering **the last question.**

To complete this phase, you can look at the diagram as many times as you need.

Once you have finished each of the phases of the experiment, please remain seated and wait for further instructions from the experiment supervisor.

PHASE 1: The Xholon Watch. Test #1

Please answer the following questions in the same order they are given. Put a "T" in the blank box with if you think that the sentence is true and an "F" if you think that it is false.

Do not forget to note down the time at which you finish answering the last question.

Thank you very much. You may begin.

1. If we are in the state TIME and the button S1 is pressed twice we reach the state TIME again.
2. There is only one possible combination of buttons to set the alarm off.
3. The chronometer may be running while the date is displayed on the watch.
4. When updating the date and time we can increase and decrease the values by pressing several buttons.
5. If button S3 is pressed for 2 s while the date is being displayed, we get to the alarm update mode.
6. The order in which the alarm, date and time are updated is always the same.
7. The diagram models how the light of the watch works.
8. While updating the alarm, date and time, the real time can be displayed at any moment by pressing one button.
9. Whenever a button is pressed, there is a transition between states.
10. There is a specific button combination to change the day of the week we are in.

TEST FINISHED IN:________minutes_________seconds

PHASE 1: The Xholon Watch. Test #2

In this phase, you will find a paragraph that describes the main features of the watch. You must complete each of the blanks in the text with a suitable word or group of words that make the text complete and meaningful.

Do not forget to write down the time after you have finished exercise.

Thank you very much. You may begin.

The Xholon DW application simulates the internal structure and the (1) of a conventional digital watch.

It displays the time or the date. By pressing the four buttons in various combinations, the current second, minute, hour, month, day and date can be updated, and various other internal functions can be managed.

It has four buttons which can be labeled S1, S2, S3 and S4. By pressing the four buttons in various combinations, the watch displays the time or the (2).

The current second, minute, hour, month, day and date can be updated, and various other internal functions can be managed, such as (3), for example.

The watch displays the date by pressing and releasing the button (4). It also allows us to change the alarm status, and there can be a beep on every (5) or when the (6) is set on. Both options may also be set on or off together.

The (7) has its typical functionalities and can be left running while the time is displayed. To change from one mode to another, button (8) must be pressed.

To update the date and time, we must press the button (9), and change the value for the alarm, time and date by pressing the button (10) to increase the displayed value in 1 unit and the button (11) to pass from one element to another.

TEST FINISHED IN: minutes seconds

PHASE 1: The Xholon Watch. Test #3

In this test, you must perform a series of tasks that will be described next. You have some blank sheets for this purpose and you can use them and bring them back at the end of this test.

When performing the required tasks, you can comment on as many aspects of your solution as you want.

Please do not forget to note down the time at which you finish carrying out the last of the tasks.

Thank you very much. You may begin.

1. Build a state machine that models the possibility of the buttons S1 and S2 being pressed together.
2. Taking as a starting point the time 00hs00mis of January 1st 2000, please indicate the minimum sequence of buttons to be pressed to update the watch to 03hs07mins of June 13th 2005.
3. What would happen if the following sequence of buttons is pressed while the watch is displaying the time?

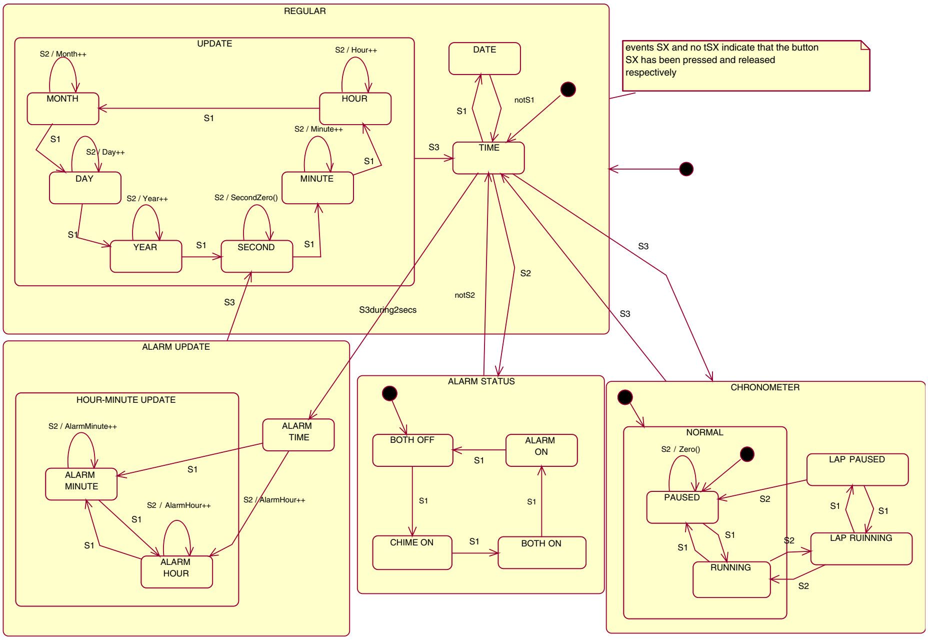
    ◆S1
    ◆S1
    ◆S2
    ◆S1
    ◆S1
    ◆S2

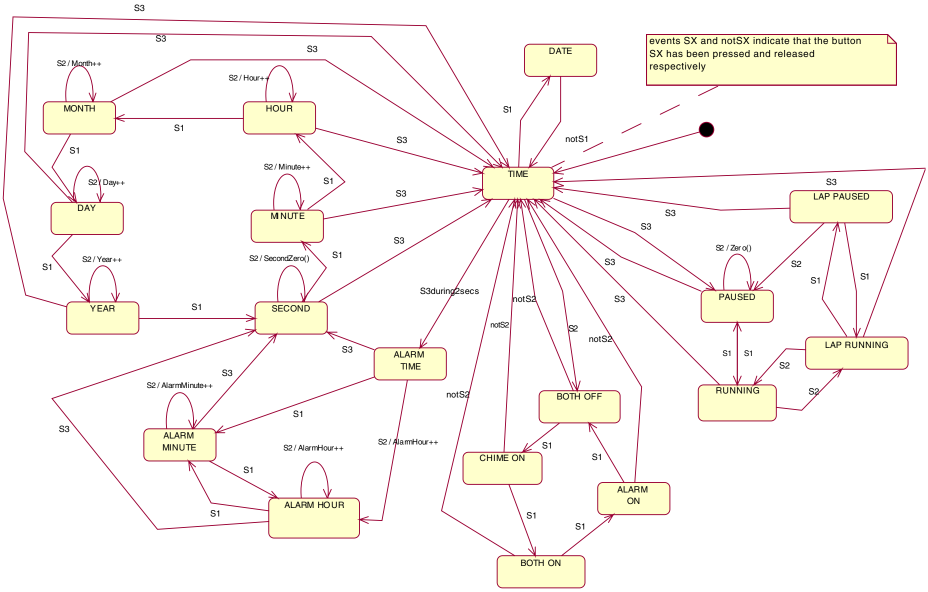
4. Please model the behavior of button S4, which turns the watch light on and off, by adding to the original diagram as many states, transitions, events, guard conditions and activities as you consider necessary.

5. Please indicate the minimum sequence of buttons to be pressed to update the alarm to 08hs00mins and activate it, but not the chime.
6. What would happen if the following sequence of buttons is pressed while the watch is displaying the time?

◆S3
◆S1
◆S2
◆S1
◆S2

TEST FINISHED IN:______minutes________seconds




events SX and notSX indicate that the button SX has been pressed and released respectively

References

Agarwal R, De P, Sinha AP (1999) Comprehending Object and Process Models: An Empirical Study. IEEE Trans Softw Eng 25(4):541–556 doi:10.1109/32.799953

Arisholm E, Sjøberg DIK (2004) Evaluating the Effect of a Delegated versus Centralized Control Style on the Maintainability of Object-Oriented Software. IEEE Trans Softw Eng 30(8):521–534 doi:10.1109/TSE.2004.43

Atkinson C, Kühne T (2003) Model Driven Development: a Metamodeling Foundation. IEEE Trans Softw Eng 20:36–41

Basili V, Shull F, Lanubile F (1999) Building Knowledge through Families of Experiments. IEEE Trans Softw Eng 25:456–473 doi:10.1109/32.799939

Biostat, Inc. (2006) Meta-Analysis v2. http://www.meta-analysis.com

Bodart F, Patel A, Sim M, Weber R (2001) Should Optimal Properties Be Used in Conceptual Modelling? A Theory and Three Empirical Tests. Inf Syst Res 12(4):384–405 doi:10.1287/isre.12.4.384.9702

Bolloju N, Leung FSK (2006) Assisting Novice Analysts in Developing Quality Conceptual Models with UML. Commun ACM 49(7):108–112 doi:10.1145/1139922.1139926

Briand L, Bunse C, Daly J (2001) A Controlled Experiment for Evaluating Quality Guidelines on the Maintainability of Object-Oriented Designs. IEEE Trans Softw Eng 27(6):513–530 doi:10.1109/32.926174

Carter J, Jaccheri L, Morasca S, Shull F (2003) Issues in Using Students in Empirical Studies in Software Engineering Education. Proc. 9th International Software Metrics Symposium (METRICS' 03), Sydney, Australia, pp 239–249

Craig SD, Gholson B, Driscoll DM (2002) Animated Pedagogical Agents in Multimedia Educational Environments: Effects of Agent Properties, Picture Features, and Redundancy. J Educ Psychol 94 (2):428–434 doi:10.1037/0022-0663.94.2.428

Cruz-Lemus JA, Genero M, Manso ME, Piattini M (2005a) Evaluating the Effect of Composite States on the Understandability of UML Statechart Diagrams. Proc. 8th International Conference on Model-Driven Engineering, Languages and Systems (MoDELS 2005), Montego Bay, Jamaica, pp 113–125

Cruz-Lemus JA, Genero M, Piattini M (2005b) Metrics for UML Statechart Diagrams. In Metrics for Software Conceptual Models. Imperial College Press, UK

Cruz-Lemus JA, Genero M, Piattini M, Toval Álvarez JA (2005c) An Empirical Study of the Nesting Level of Composite States within UML Statechart Diagrams. Proc. First International Workshop on Best Practices of UML (BP-UML 2005)-ER 2005 Workshops, Klagenfurt (Austria), pp 12–22

Denger C, Ciolkowski M (2003) High Quality Statecharts through Tailored, Perspective-Based Inspections. Proc. 29th EUROMICRO Conference "New Waves in System Architecture", Belek, Turkey, pp 316–325

Dybå T, Arisholm E, Sjøberg DIK, Hannay JE, Shull F (2007) Are Two Heads Better than One? On the Effectiveness of Pair Programming. IEEE Softw 24(6):10–13 doi:10.1109/MS.2007.158

Fenton N, Pfleeger S (1997) Software Metrics: A Rigurous and Practical Approach. International Thompson Publishing Inc, London

Firesmith D, Henderson-Sellers B, Graham I (1998) OPEN modeling language (OML) reference manual. Cambridge University Press, New York, USA

Gemino A, Wand Y (2003) Evaluating Modeling Techniques based on Models of Learning. Commun ACM 46(10):79–84 doi:10.1145/944217.944243

Gemino A, Wand Y (2005) Complexity and Clarity in Conceptual Modeling: Comparison of Mandatory and Optional Properties. Data Knowl Eng 55:301–326 doi:10.1016/j.datak.2004.12.009

Genero M, Manso ME, Visaggio A, Piattini M, Canfora G (2007) Building Measure-based Prediction Models for UML Class Diagram Maintainability. Empir Softw Eng 12(5):527–549 doi:10.1007/s10664-007-9038-4

Genero M, Cruz-Lemus JA, Caivano D, Abrahao S, Insfran E, Carsí JA (2008) Assessing the Influence of Stereotypes on the Comprehension of UML Sequence Diagrams: A Controlled Experiment. Lecture Notes in Computer Science (5301): 11th ACM/IEEE MODELS Conference, 280–294, October

Glass GV, McGaw B, Smith ML (1981) Meta-Analysis in Social Research. Sage Publications.

Glezer C, Last M, Nachmany E, Shoval P (2005) Quality and Comprehension of UML Interaction Diagrams—An Experimental Comparison. Inf Softw Technol 47:675–692 doi:10.1016/j.infsof.2005.01.003

Harrison R, Counsell S, Nithi R (2000) Experimental Assessment of the Effect of Inheritance on the Maintainability of Object-Oriented Systems. J Syst Softw 52:173–179 doi:10.1016/S0164-1212(99)00144-2

Hayes W (1999) Research in Software Engineering: a Case for Meta-Analysis. Proc. 6th IEEE International Symposium on Software Metrics (METRICS' 99), Boca Raton, USA, pp 143–151

Hedges LV, Olkin I (1985) Statistical Methods for Meta-Analysis. Academia Press.

Höst M, Regnell B, Wohlin C (2000) Using Students as Subjects—a Comparative Study of Students & Profesionals in Lead-Time Impact Assessment. Proc. 4th Conference on Empirical Assessment & Evaluation in Software Engineering (EASE 2000), Keele, UK, pp 201–214

Hu Z, Shatz SM (2006) Explicit Modeling of Semantics Associated with Composite States in UML Statecharts. Autom Softw Eng 13(4):423–467 doi:10.1007/s10515-006-0272-6

ISO/IEC (2001) Software Product Evaluation-Quality Characteristics and Guidelines for their Use. ISO/IEC Standard 9126.

Juristo N, Moreno A (2001) Basics of Software Engineering Experimentation. Kluwer Academic Publishers.

Kampenes V, Dybå T, Hannay JE, Sjoberg DIK (2007) A Systematic Review of Effect Size in Software Engineering Experiments. Inf Softw Technol 49(11–12):1073–1086 doi:10.1016/j.infsof.2007.02.015

Khatri V, Vessey I, Ramesh PCV, Park SJ (2006) Understanding Conceptual Schemas: Exploring the Role of Application and IS Domain Knowledge. Inf Syst Res 17(1):81–99 doi:10.1287/isre.1060.0081

Kirk RE (1995) Experimental Design: Procedures for the Behavioral Sciences. Brooks/Cole Publishing Company.

Kitchenham B, Pfleeger S, Pickard L, Jones P, Hoaglin D, El-Emam K, Rosenberg J (2002) Preliminary Guidelines for Empirical Research in Software Engineering. IEEE Trans Softw Eng 28(8):721–734 doi:10.1109/TSE.2002.1027796

Laitenberger O, El-Emam K, Harbich T (1999) An Internally Replicated Quasy-Experimental Comparison of Checklist and Perspective-based Reading of Code Documents. 006.99/e. IESE.

Lipsey M, Wilson D (2001) Practical Meta-Analysis, Sage.

Miller J (2000) Applying Meta-Analytical Procedures to Software Engineering Experiments. J Syst Softw 54:29–39 doi:10.1016/S0164-1212(00)00024-8

Pickard LM (2004) Combining Empirical Results in Software Engineering, Univer-sity of Keele, T-R V1.

Mayer RE (1989) Models for Understanding. Rev Educ Res 59(1):43–64

Mayer RE, Anderson RB (1991) Animations need Narrations: An Experimental Test of the Dual-Coding Hypothesis. J Educ Psychol 83(4):484–490 doi:10.1037/0022-0663.83.4.484

Mayer RE (2001) Multimedia Learning. Cambridge University Press.

Miller J, McDonald F (1998) Statistical Analysis of Two Experimental Studies. EFoCS-31-98. University of Strathclyde.

OMG (2003) UML 2.0-2nd Revised Submission. document ptc/03-01-07. Object Management Group.

Otero MC, Dolado JJ (2004) Evaluation of the Comprehension of the Dynamic Modeling in UML. Inf Softw Technol 46(1):35–53 doi:10.1016/S0950-5849(03)00108-3

Otero MC, Dolado JJ (2005) An Empirical Comparison of the Dynamic Modeling in OML and UML. J Syst Softw 77(2):91–102 doi:10.1016/j.jss.2004.11.022

Pigoski T (1997) Practical Software Maintenance. New York, USA: Wiley Computer Publishing

Porter A, Johnson M (1997) Assessing Software Review Measurement: Necessary and Sufficient Properties for Software Measures. Inf Softw Technol 42(1):35–46

Purchase HC, Colpoys L, McGill M, Carrington D, Britton C (2001) UML Class Diagram Syntax: an Empirical Study of Comprehension. Proc. Australian Symposium on Information Visualisation, Sydney, Australia, pp 113–120

Purchase HC, Colpoys L, McGill M, Carrington D (2002) UML Collaboration Diagram Syntax: an Empirical Study of Comprehension. Proc. 1st International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT'02), Paris, France, pp 13–22

Ricca F, Di Penta M, Torchiano M, Tonella P, Ceccato M (2007) The Role of Experience and Ability in Comprehension Tasks supported by UML Stereotypes. Proc. 29th International Conference on Software Engineering (ICSE, 07), Minneapolis, USA, pp 375–384

Rosenthal R (1986) Meta-Analytic Procedures for Social Research. Sage Publications.

Rosenthal R (1994) Parametric measures of effect size. In The Handbook of Research Synthesis. Russell Sage Foundation, New York

Selic B (2003) The Pragmatics of Model-Driven Development. IEEE Softw 20(5):19–25 doi:10.1109/MS.2003.1231146

Sjoberg DIK, Hannay JE, Hansen O, Kampenes V, Karahasanovic A, Liborg NK, Rekdal AC (2005) A Survey of Controlled Experiments in Software Engineering. IEEE Trans Softw Eng 31(9):733–753 doi:10.1109/TSE.2005.97

SPSS (2003) SPSS 12.0, Syntax Reference Guide. SPSS Inc, Chicago, USA

Staron M, Kuzniarz L, Wohlin C (2006) Empirical Assessment of Using Stereotypes to Improve Comprehension of UML Models: a Set of Experiments. J Syst Softw 79:727–742 doi:10.1016/j.jss.2005.09.014

Sutton JA, Abrams RK, Jones RD, Sheldon AT, Song F (2001) Methods for Meta-Analysis in Medical Research. John-Wiley & Sons.

Tabbers HK (2004) Multimedia Instructions and Cognitive Load Theory: Effects of Modality and Cueing. Br J Educ Psychol 74(1):71–81 doi:10.1348/000709904322848824

Thomas D (2004) MDA: Revenge of the Modelers or UML Utopia. IEEE Softw 21(3):15–17 doi:10.1109/MS.2004.1293067

Tichy WF (2000) Hints for Reviewing Empirical Work in Software Engineering. Empir Softw Eng 5:309–312 doi:10.1023/A:1009844119158

Verelst J (2004) The Influence of the Level of Abstraction on the Evolvability of Conceptual Models of Information Systems. Proc. 3rd International Symposium on Empirical Software Engineering (ISESE 2004), Redondo Beach, USA, pp 17–26

Webb K (2006) Xholon Digital Watch Project. http://www.primordion.com/Xholon/samples/watch.htm

Winer BJ, Brown DR, Michels KM (1991) Statistical Principles in Experimental Design. McGraw-Hill.

Wohlin C, Runeson P, Hast M, Ohlsson MC, Regnell B, Wesslen A (2000) Experimentation in Software Engineering: an Introduction. Kluwer Academic Publisher.

Wolf FM (1986) Meta-Analysis: Quantitative Methods for Research Synthesis. Sage Publications.

Woodfield SN, Dunsmore HE, Shen VY (1981) The Effect of Modularization and Comments on Program Comprehension. Proc. 5th International Conference on Software Engineering (ICSE 1981), San Diego, USA, pp 215–223

Xie S, Kraemer E, Stirewalt REK (2007) Empirical Evaluation of a UML Sequence Diagram with Adornments to Support Understanding of Thread Interactions. Proc. 15th IEEE International Conference on Program Comprehension (ICPC'07), Banff, Canada, pp 123–134

Yusuf S, Kagdi H, Maletic JI (2007) Assessing the Comprehension of UML Class Diagrams via Eye Tracking. Proc. 15th IEEE International Conference on Program Comprehension (ICPC'07), Banff, Canada, pp 113–122

José A. Cruz-Lemus is an Assistant Professor at the Department of Information Systems and Technologies at the University of Castilla-La Mancha, Ciudad Real, Spain. He is MSc and PhD in Computer Science from the same university. His main research interests are empirical software engineering, software metrics and UML models quality. He has published his works in several journals (Empirical Software Engineering, Information Sciences, and International Journal of Engineering Education), conferences (MoDELS, E/R, ESEM, etc.), and several book chapters.

Marcela Genero is Associate Professor at the Department of Information Systems and Technologies at the University of Castilla-La Mancha, Ciudad Real, Spain. She received her MSc degree in Computer Science in the Department of Computer Science of the University of South, Argentine in 1989, and her PhD at the University of Castilla-La Mancha, Ciudad Real, Spain in 2002. Her research interests are: empirical software engineering, software metrics, UML models quality, quality in product lines, quality in MDD, etc. Marcela Genero has published in prestigious journals (Journal of Software Maintenance and Evolution: Research and Practice, Data and Knowledge Engineering, Empirical Software Engineering, Information and Software Technology, etc.), and conferences (CAISE, E/R, MODELS/UML, ESEM, METRICS, etc. She has participated in several program committees (CAISE 2005, METRICS, 2004-2005-2006, ICSM 2007-2008, ESEM 2008-2009, etc.). She is member of the International Software Engineering Research Network (ISERN).

M. Esperanza Manso is T.E.U. Professor at the department of Computer Language and Systems University of Valladolid, Spain. She has a major in Mathematics from the University of Valladolid. Currently she is working towards her PhD. Her main research interests are software maintenance, reengineering and reuse experimentation. Her works have been accepted in national and international congress (JISBD, OOIS, CAISE, METRICS, ISESE, etc.), international journals (Empirical Software Engineering) and several works have been accepted as book chapters.

Sandro Morasca is a Professor of Computer Science at the Università degli Studi dell'Insubria in Como, Italy. In the past, he was an Associate Professor and Assistant Professor at the Politecnico di Milano in Milano, Italy. He was a Faculty Research Assistant and later a Visiting Scientist at the Department of Computer Science of the University of Maryland at College Park. Sandro Morasca has been actively carrying out research in the Software Engineering field in Empirical Software Engineering, Specification of Concurrent and Real-time Software Systems, Software Verification, and Open Source Software, and has published about 20 journal papers and 50 conference papers. Sandro Morasca has been involved in a number of national and international projects. He is currently the Leader of the activity related to the trustworthiness of Open Source Software in the QualiPSo project, financed by the European Union. Sandro Morasca has served on the PC of a number of international software engineering conferences and serves on the Steering Committee of the international conference "ESEM" and on the editorial board of "Empirical Software Engineering: An International Journal," published by Springer-Verlag.

Mario Piattini has an MSc and PhD in Computer Science from the Technical University of Madrid and is a Certified Information System Auditor and Certified Information Security Manager by ISACA (Information System Audit and Control Association). He is a professor in the Department of Computer Science at the University of Castilla-La Mancha, in Ciudad Real, Spain. Author of several books and papers on software engineering, databases and information systems, he leads the ALARCOS research group of the Department of Information Systems and Technologies at the University of Castilla-La Mancha, in Ciudad Real, Spain.