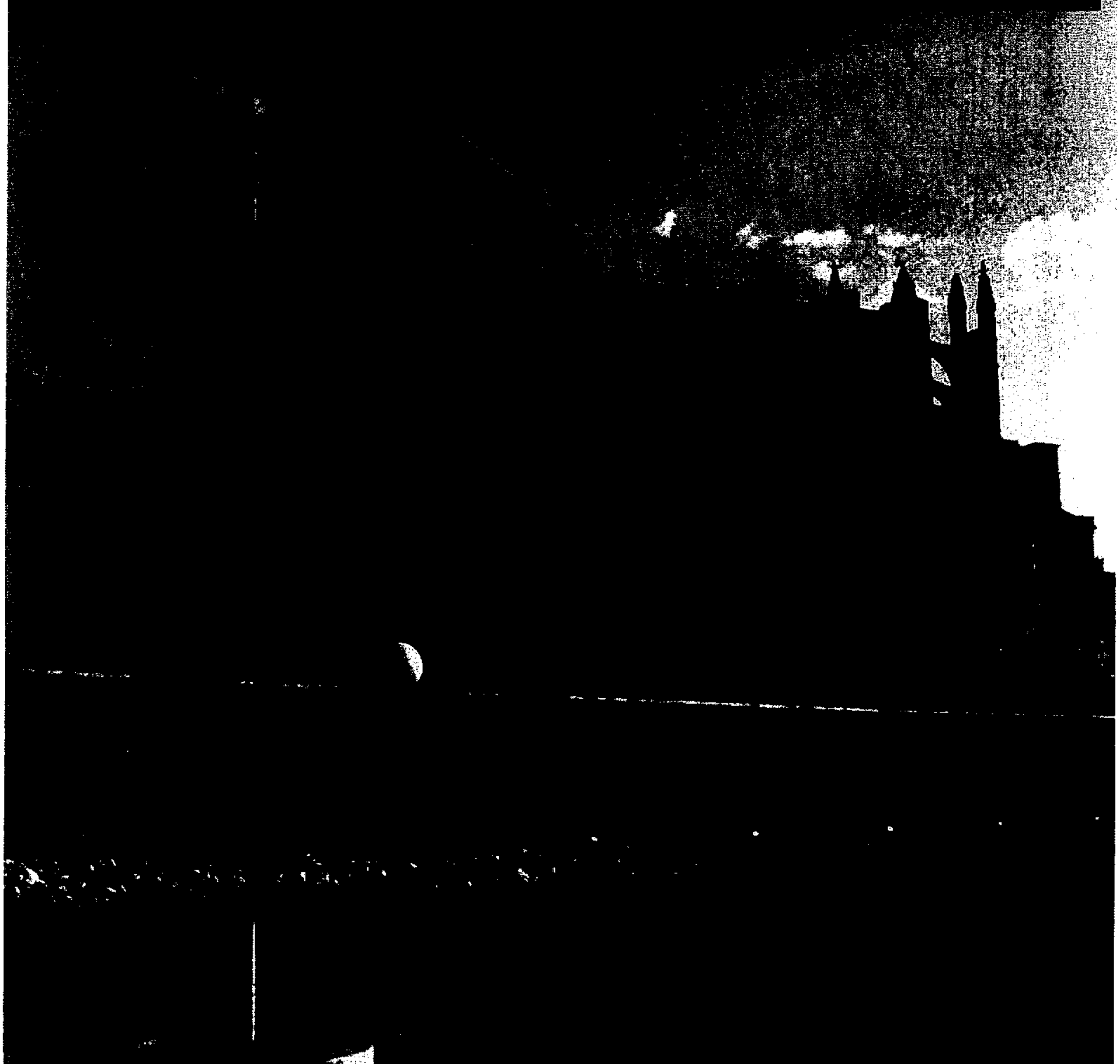
Proceedings of the

IWSM-Mensura 2007

International Conference on Software Process and Product Measurement



Palma de Mallorca, Spain

Editors

Alain Abran
Department of Software and IT Engineering
École de technologie supérieure
1100, Notre-Dame Street West
Montréal, Québec
Canada H3C 1K3
E-mail: Alain.Abran@etsmtl.ca

Reiner Dumke
Otto von Guericke University of Magdeburg
Postfach 4120
39016 Magdeburg
Germany
E-mail: dumke@ivs.cs.uni-magdeburg.de

Antonia Mas
Research Group for Software Process Improvement
Department of Computer Sciences and Mathematics
University of the Balearic Islands
Escola Politècnica Superior
Edifici A. Turmeda. Campus universitari
Cra. de Valldemossa, Km. 7.5
07122 Palma de Mallorca (Illes Balears)
Spain
E-mail: antonia.mas@uib.es

Edition: Edicions UIB
Campus de la Universitat de les Illes Balears
Cra. de Valldemossa, km 7.5
07122 – Palma (Illes Balears)
http://edicions.uib.es

ISBN: 978-84-8384-020-7
Dipòsit Legal: PM 2625-2007

Alain Abran
Reiner Dumke
Antonia Mas (Eds.)

PROCEEDINGS OF THE IWSM - MENSURA 2007

INTERNATIONAL WORKSHOP ON SOFTWARE MEASUREMENT - INTERNATIONAL CONFERENCE ON SOFTWARE PROCESS AND PRODUCT MEASUREMENT

PALMA DE MALLORCA – SPAIN
NOVEMBER, 5 – 8, 2007

Preface

This volume contents the papers presented at the IWSM-Mensura 2007 conference held in Palma de Mallorca, Spain in November 2007.

IWSM-Mensura 2007 is a continuation of two international conferences IWSM (International Workshop in Software Measurement) and Mensura (International Conference on Software Process and Product Measurement). They are conducted jointly for the first time.

The objective of the IWSM-Mensura 2007 conference is to bring to light the most recent findings and results in the area of software measurement and to stimulate discussion between researchers and professionals.

We would like to thank the many people who have brought this International Conference into being. On the one hand, we want to express our thanks to the Organizing Committee members who have made possible that all conference events, both technical and social, have been performed successfully.

On the other hand, we want to appreciate the work done by all the people that have participated in the organization of sessions: special sessions, workshops and industrial track. Finally, we also want to thank the Program Committee Members for their hard work in reviewing both the abstracts and the final papers.

The organizers would also like to thank the University of the Balearic Islands, the Research Group for Software Process Improvement and the Government of the Balearic Islands for supporting the conference.

November 2007

Reiner Dumke and Miguel Toro
Alain Abran and Antonia Mas
General and Program Chairs

Conference Organization

General Chairs

Reiner Dumke, *University of Magdeburg, Germany*
Miguel Toro, *University of Seville, Spain*

Program Committee Chairs

Antonia Mas, *University of the Balearic Islands, Spain*
Alain Abran, *University of Québec, Canada*

Proceedings Chairs

Daniel Rodríguez, *University of Alcalá, Spain*
Maya Daneva, *Univerity of Twenty, The Netherlands*

Program Committee Members

Moteoi Azuma, *Japan*
Manfred Bundschuh, *DASMA, Germany*
David Card, *USA*
François Coallier, *ÉTS, Canada*
Juan J. Cuadrado-Gallego, *University of Alcala, Spain*
Jean-Marc Desharnais, *ÉTS, Canada*
Ton Dekkers, *Shell Information Technology, Netherlands*
Javier Dolado, *University of the Basque Country, Spain*
Christof Ebert, *Vector Consulting, Germany*
Khaled El Emam, *University of Ottawa, Canada*
Juan Garbajosa, *U. Politécnica, Madrid, Spain*
Felix García, *University of Castilla-La Mancha, Spain*
Elena García-Barriocanal, *University of Alcalá, Spain*

George Ghinea, *Brunel Univesity, UK*

Naji Habra, *FUNDP, Namur, Belgium*

Nadine Hanebutte, *University of Idaho, USA*

Mark Harman, *Kings College, UK*

Rachel Harrison, *UK*

Ali Idri, *INSIAS, Morocco*

Natalia Juristo, *U. Politécnica, Madrid, Spain*

Adel Khelifi, *U. Al Hosn, URA*

Pedro Lara, *University Europea, Madrid, Spain*

Miguel Lopez, *Belgium*

Mathias Lother, *Germany*

Hakim Lounis, *Canada*

Fernando Machado, *Catholic University of Uruguay, Uruguay*

John McGarry, *USA*

Roberto Meli, *DPO, Italy*

Pam Morris, *Australia*

John C. Munson, *USA*

Olga Ormandjieva, *Concordia University, Canada*

Oscar Pastor, *Technical University of Valence, Spain*

Mario Piattini, *University of Castilla-La Mancha, Spain*

Tony Rollo, *UK*

Salvador Sánchez, *University of Alcalá, Spain*

Miguel-Ángel Sicilia, *University of Alcalá, Spain*

Manoranjan Satpathy, *General Motors, India*

Andreas Schmietendorf, *FH Harz Wernigerode, Germany*

Manuel Serrano, *University of Castilla-La Mancha, Spain*

Marcio Silveira, *EDS, Brazil*

Harry Sneed, *SES Munich/Budapest, Hungary*

Andreas Schmientendorf, *FHW Berlin, Germany*

Esperança Amengual, *University of the Balearic Islands, Spain*

Stanimir Stojanov, *Bulgaria*

Hannu Toivonen, *Nokia, Finland*

Claes Wohlin, *Blekinge Institute of Technology, Sweden*
David Zubrow, *USA*
Horst Zuse, *TU Berlin, Germany*
Marcus Ciolkowski, *Fraunhofer IESE, Germany*
Javier Aroba, *University of Huelva, Spain*
Dietmar Pfalh, *University of Calgary, Canada*
Andreas Jedlitschka, *Fraunhofer IESE, Germany*

Organizing Committee

Antonia Mas, *University of the Balearic Islands, Spain*
Esperança Amengual, *University of the Balearic Islands, Spain*
Ana Belén Petro Balaguer, *University of the Balearic Islands, Spain*
Carlos Guerrero Tomé, *University of the Balearic Islands, Spain*
Antoni Lluís Mesquida Calafat, *University of the Balearic Islands, Spain*

Special Sessions/Workshops Chairs

Mercedes Ruíz, *University of Cádiz, Spain*
Luis Fernández, *University Europea de Madrid, Spain*

Industrial Track Chairs

Luigi Buglione, *AtosOrigin, Italy*

Industrial Track Program Committee

Alfonso Romo, *El Corte Inglés, Spain*
Victoria de MENA, *El Corte Inglés, Spain*
Carmen Velasco, *El Corte Inglés, Spain*
Miguel Lopez, *Algorismi, Belgium*
Luca Santillo, *Independent consultant, Italy*
Sylvie Trudel , *CRIM, Canada*

Index

Michael Berry and Chris S. Johnson 1

Improving the Quality of Information for Software Project Management

Ayaz Farook and Reiner R. Dumke 11

Developing and Applying a Consolidated Evaluation Framework to Analyze Test Process Improvement Approaches

Rafik Ounanouki and Alain April 26

IT Process Conformance Measurement: A Sarbanes-Oxley Requirement

Ayça Tarhan and Onur Demirors 38

Assessment of Software Process and Metrics to Support Quantitative Understanding

Ali Idri, Emilia Mendes, Abdelali Zakrani 48

Web Cost Estimation Models using Radial Basis Function Neural Networks

Emilia Mendes, Sergio Di Martino, Filomena Ferrucci, Carmine Gravino, Steve Counsell 58

Comparing Machine-learning Techniques for Web Cost Estimation

José Ignacio Panach, Nelly Condori-Fernández, Francisco Valverde, Nathalie Aquino, Óscar Pastor 67

Towards an Early Usability Evaluation for Web Applications

Stamatia Bibi, Nikolaos Mittas, Lefteris Angelis, Ioannis Stamelos, Emilia Mendes 77

Comparing Cross- vs. Within-Company Effort Estimation Models Using Interval Estimates

Francisco Valdés Souto, Alain Abran 87

Industry Case Studies of Estimation Models Using Fuzzy Sets

M. Garre, J. J. Cuadrado, M. A. Sicilia, I. Sánchez 102

Analysis of Heterogeneous Software Projects Databases

Gemma Grau, Xavier Franch 110

Using the PriM method to Evaluate Requirements Models with COSMIC-FFP

Yoshiki Mitani, Tomoko Matsumura, Mike Barker, Seishiro Tsuruho, Katsuro Inoue, Ken-Ichi Matsumoto 121

An Empirical Study of Process Management and Metrics based on In-process Measurements of a Standardized Requirements Definition Phase

Manar Abu Talib, Adel Khelifi, Alain Abran, Olga Ormandjieva 132

A case study using the COSMIC-FFP Measurement Method for Assessing Real-Time System Specifications

Harold van Heeringen 142

Speeding up the estimation process with the Estimating Wizard

Frank Vogelezang 153

Scope Management – How Uncertain is Your Certainty

Andrea Raffaelli, Loredana Mancini 171

Impact of Improvement Actions on Help Desk Costs

Maya Daneva 182

Preliminary Results in a Multi-site Empirical Study on Cross-organizational ERP Size and Effort Estimation

María Díaz Ley, Félix García, Mario Piattini 194
Implementing Software Measurement Programs in Non mature Small Settings

Jari Soini, Vesa Tenhunen, Markku Tukiainen 204
Software Metrics and Evaluation of Their Uselfulness in Finnish Software Companies

Alain Abran, Juan Garbajosa, Laila Cheikhi 216
Estimating the Test Volume and Effort for Testing and Verification & Validation

Cigdem Gencel, Luigi Buglione 235
Do Different Functionality Types Affect the Relationship between Software Functional Size and Effort?

M. Kassab, O. Ormandjieva, M. Daneva, A. Abran 247
Non Functional Requirements: Size Measurement and Testing with COSMIC-FFP

Bruno Rossi, Barbara Russo, Giancarlo Succi 260
A Method to Measure Software Adoption in Organizations: A preliminary Study

José A. Cruz-Lemus, Marcela Genero, Mario Piattini 269
Using Controlled Experiments for Validating UML Statechart Diagrams Measures

Dominik Gessenharter, Alexander-Marc Merten, Alexander Raschke, Nicolás Fernando Porta 279
Experiences on Using Software Experiments in the Validation of Industrial Research Questions

Andreas Jedlitschka 289
An Infraestructure for Empirically-based Software Engineering Technology Selection

Maurizio Morisio, Evgenia Egorova, Marco Torchiano 299
Why Software Projects fail? Empirical Evidence and Relevant Metrics

Tilmann Hampp 309
A Model of Costs and Benefits of Reviews

Martin Kunz, Reiner R. Dumke, René Braungarten, Andreas Schmietendorf
How to measure Agile Software Developement 319

Miguel López, Naji Habra 326
Towards a Two-dimensional Approach To track Software Degradation

Using Controlled Experiments for Validating UML Statechart Diagrams Measures

José A. Cruz-Lemus, Marcela Genero and Mario Piattini

ALARCOS Research Group
Department of Information Technologies and Systems
Indra-UCLM Research and Development Institute
University of Castilla-La Mancha
Paseo de la Universidad, 4 – 13071 Ciudad Real, Spain
{JoseAntonio.Cruz, Marcela.Genero, Mario.Piattini}@uclm.es

Abstract. In this work, we present the main conclusions obtained from the definition and validation of a set of measures for UML statechart diagrams, in a methodological way. The main focus is the empirical validation of the measures as early understandability indicators.

1. Introduction

The availability of valid measures in the early phases of the software development life-cycle allows a better management of the later phases. The measures allow the designers a quantitative comparison of design alternatives, and therefore an objective selection among several conceptual modelling alternatives with equivalent semantic content. Besides, designers can predict external quality characteristics, like maintainability, in the initial phases of the software life-cycle and better allocate the resources based on these predictions.

In order to define valid measures, we have followed and refined a process for measure definition [7] that consists of three main steps: measure definition, and theoretical and empirical validation. This process pays especial emphasis on some issues that must be taken into account when defining measures for software, such as:

- Measures must be well-defined, pursuing clear goals.
- Measures must be theoretically validated, by addressing the question 'is the measure measuring the attribute it is purporting to measure?'
- Measures must be empirically validated, by addressing the question 'is the measure useful in the sense that it is related to other external quality attributes in the expected way?'
- Measures must be defined in a precise way, avoiding misconceptions and misunderstandings.
- Measures calculation must be easy and it is better if their extraction is automated by a tool.

In this work, we present the main results obtained after defining a set of measures keeping these issues in mind, paying special attention to the empirical validation of the measures.

Section 2 presents the informal and formal definition of the measures. Section 3 tackles their theoretical validation. Section 4 explains the different families of experiments performed for achieving a thorough empirical validation of the measures. Section 5 provides the main features of a tool developed for the automatic calculation of the measures. Finally, section 6 summarizes the main conclusions achieved and outlines the future work.

2 Measures Definition

The main concern of this research was the definition of a set of early indicators of the understandability of UML statechart diagrams. But understandability, as an external quality attribute, is hard to measure early in the modelling process. Therefore, an indirect measurement based on internal properties of the model such as the structural complexity, was required [6].

The main concern of the measures definition step is explaining what the measures intend to measure. In the recent years, a great number of measures proposals for OO software products have been developed but most of them present a lack of formalization in their definition. This fact leads a set of difficulties to arise such as [1]:

- Experimental findings can be misunderstood due to the fact may be not clear what the measure really captures.
- Measures extraction tools can arrive to different results.
- Experiments replication is hampered.

Our work has refined the referred method by formally defining the measures using two formal languages: OCL [17] and Maude [9]. Table 1 presents the informal definition of the measures in natural language whilst and Fig. 1 provides an example of the formal definition of the measure NCS using OCL.

Table 1. Definition of the measures in natural language

Nesting Level in Composite States	NLCS	The maximum number of composite states nested within other composite states in the statechart diagram.
Number of Activities	NA	The total number of activities in the statechart diagram.
Number of Composite States	NCS	The total number of composite states in the statechart diagram.
Number of Complex Transitions	NCT	A complex transition has attached a number of guard conditions, events or actions, while a simple transition does not.
Number of Events	NE	The total number of events, whichever the type they are.
Number of Entry Actions	NEntryA	The total number of entry actions, i.e., the actions performed each time a certain state is reached.
Number of Exit Actions	NExitA	The total number of exit actions, i.e., the actions performed each time a certain state is left.
Number of Guards	NG	The total number of guard conditions of the statechart diagram.
Number of Indirect Actions	NIA	Number of actions to be performed associated to transitions.
Number of Simple States	NSS	The total number of states, also considering the simple states within the composite states.

Number of Transitions	NT	The total number of transitions, considering common transitions, the initial and final transitions, self-transitions and internal transitions

```
context StateMachine::NCS():Integer
body: result = self.top.allSubstates()->collect(s |
               s.oclIsTypeof(CompositeState))-> size()
```

Fig. 1. Formal definition of the NCS measure using OCL

3 Theoretical Validation

The theoretical validation was carried out to show that a measure is really measuring the attribute that it aims to measure [2]. Moreover, it provides information related to the mathematical and statistical operations that can be done with the measures, e.g., the scale in which a measure should be measured.

In our work, we have based on the property-based framework proposed by Briand [3, 4] and the Measurement Theory-based DISTANCE framework [18]. The first one has characterized the measures as size or complexity measures, while the second framework characterized all the measures as ratio scale.

4 Empirical Validation

Empirical validation is an on-going activity [2] performed to demonstrate the usefulness of a measure. This phase is necessary before any attempt to use measures as objective and early indicators of quality.

In this section, we will describe the three families of experiments that have been carried out during our work. In section 4.1, we will explain a family of three experiments performed in order to empirically validate the measures proposed in this PhD thesis and build a preliminary understandability prediction model by means of a regression analysis using a technique specifically recommended when the data had been obtained through a repeated measures design.

The second empirical study, described in section 4.2, is another family composed by five different empirical studies. This family was used for assessing how composite states affected the understandability of UML statechart diagrams.

The third family, explained in section 4.3, is composed by a controlled experiment and a replication of it that were performed in order to study the optimal nesting level of composite states within the UML statechart diagrams.

Fig. 2 gives a global vision of the whole empirical validation process.

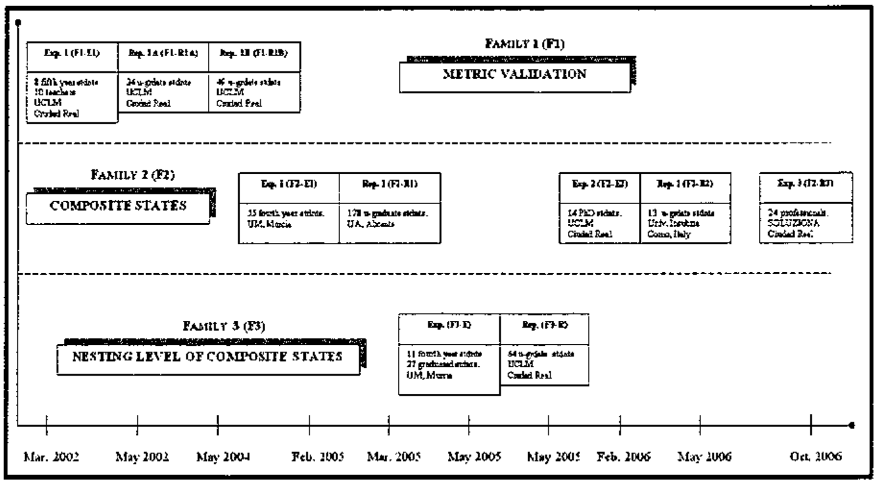


Fig. 2. Chronology of the families of experiments

4.1 First family of experiments (F1): Performing the measures validation

This first family of experiments was performed for studying the relationship between the different defined measures and the understandability of UML statechart diagrams. The main characteristics of this family can be found in Table 2, while Fig.3 graphically shows the average values that were obtained for the different measurements related to the dependent variable.

Table 2. Characteristics of the family F1

	Subjects	E1: 8 PhD students, 10 teachers R1A: 24 students R1B: 49 students
	Location	University of Castilla-La Mancha (Spain)
	Date	E1: March 2002 R1A: May 2002 R1B: May 2004
		20 diagrams with different values for the measures
	Dependent	Understandability of UML statechart diagrams, measured by UT (time), UCorr (correctness) and UCom (completeness), later by UEffic (efficiency).
	Independent	Measures for UML statechart diagrams understandability
		$H_{0,1}$: There is not a significant correlation between the UML statechart diagrams measures and the understandability time. $H_{0,2}$: There is not a significant correlation between the UML statechart diagrams size measures and understandability correctness. $H_{0,3}$: There is not a significant correlation between the UML statechart diagrams size measures and understandability completeness.

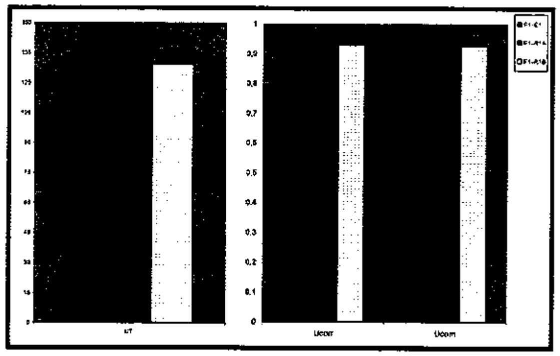


Fig. 3. Average values for UT, UCorr and UCom in F1

We could observe that while for UCorr and UCom, the values seemed quite logical and the teachers and higher-degree students had obtained better results, we did not obtain those results with the UT. This fact made us think that time, on its own, was not a good indicator of the understandability of the diagrams, so we used the understandability efficiency, which relates the number of correct answers with the time invested answering the questions. The results with this new measure agreed with those obtained for UCorr and UCom.

Later, we performed a Spearman's correlation analysis and obtained that the measures NA, NSS, NG and NT were highly correlated with the understandability efficiency of the diagrams.

We also performed a PCA that characterized the measures into three different groups:

- Simple States Features (SSF), composed by measures that explore the relationships between the different states of the diagrams and also the states themselves.
- Activities within States (AWS), composed by the measures of the activities performed after entering or leaving a state.
- Number of Activities (NA), composed only by this measure.

Finally we built the preliminary regression model for the understandability efficiency shown next:

$$UEff = 0.011575 - 0.000813*NA - 0.000204*SSF - 0.000273*AWS \quad (1)$$

More details about this first family of experiments can be found in [13]

4.2 Second family of experiments (F2): Studying the composite states

In the previous study, the composite states did not show a clear effect on the understandability of the diagrams, so we decided to study them specifically. The main characteristics to the different studies performed in this family are shown in Table 3.

Table 3. Characteristics of the family F2

[redacted]	**Subjects**	E1: 55 students R1:178 students E2: 14 PhD students R2: 13 students E3: 24 professionals
	Location	E1: University of Murcia R1: University of Alicante E2: University of Castilla-La Mancha R2: Università dell'Insubria E3: SOLUZIONA SW-Factory
	Date	E1: February 2005 R1: March 2005 E2:February 2006 R2: May 2006 E3: October 2006
	Dependent	Understandability of UML statechart diagrams, measured by UEffec (always) and UReten and UTrans (E2, R2 & E3).
	Independents	The use of composite states in the diagram (always) and the domain of the diagram (E1, R1, E2 & R2).
	[redacted]	H1a: using composite states improves UEffec in subjects when trying to understand an UML statechart diagram. (always) H1b: using composite states improves UTrans in subjects when trying to understand an UML statechart diagram. (E2, R2 & E3) H1c: using composite states improves UReten in subjects when trying to understand an UML statechart diagram. (E2, R2 & E3)

The main strength of this family relies on the evolution of materials, tasks, and subjects that it has suffered along its performance.

First, we noticed that the materials used were not complicated enough to obtain actual results, so we increased their difficulty from the different experiments until using a real-project model in the last experiment (E3) [21].

The original tasks also evolved by the use of the Cognitive Theory of Multimedia Learning [16], which provided the measures of transfer and retention for measuring the dependent variable.

Finally, we used students in the first studies but in the last one, we counted on a set of real practitioners in order to alleviate the possible lack of experience that the students might have.

The conclusions of this family indicate that the use of composite states does not significantly improve the understandability of UML statechart diagrams, at least when working with diagrams whose size and complexity are not too high.

More details about this second family of experiments can be found in [11, 12].

4.3 Third family of experiments (F3): Looking for the nesting level of composite states

The use of hierarchical structures and how they affected the quality of different modeling techniques has been broadly studied [5, 8, 14, 19, 20]. In the same direction that most of these works, we intended to assess which was the optimal level of inheritance within a composite state in an UML statechart diagram. This was the aim of this third family of experiments that we performed. Its main characteristics are detailed in Table 4.

Table 4. Characteristics of the family F3

[redacted]	**Subjects**	Exp: 38 students Rep: 64 students
	Location	Exp: University of Murcia Rep: University of Castilla-La Mancha
	Date	May 2005
	[redacted]	3 diagrams with values 0, 1 and 2 for the measure NLCS
	Dependent	Understandability of UML SD, measured by UCorr and UEffic.
	Independent	Nesting Level within composite states in an UML SD
	[redacted]	$H_{0\text{-}ij}$: the understandability of UML statechart diagrams with i and j composite states nesting levels is not significantly different $H_{1\text{-}ij}$: the understandability of UML statechart diagrams with i and j composite states nesting levels is significantly different In both cases, $i, j \in \{0, 1, 2\}$ and $i \neq j$.

The results obtained indicating that a flat nesting level within composite states made the diagrams more understandable.

More details about this third family of experiments can be found in [10].

5 GenMETRIC

GenMETRIC [15] is a tool for defining, calculating and visualizing software measures. This tool supports the management of the measurement process by supporting the definition of measurement models, the calculation of the measures defined in those measurement models and the presentation of the results in tables and graphically.

The two key characteristics of GenMETRIC are:

- Genericity. With this tool it is possible to measure any software entity. The requirement necessary to achieve this goal is that the metamodel representing the software entity (domain metamodel) must be included in the repository of the tool. The different measures must be defined on the elements of the domain metamodels. This implies that in order to measure new entities it is not necessary to add a new code to GenMETRIC.
- Extensibility. GenMETRIC supports the definition of any software measure. The base measures are defined on the domain metamodel elements (classes and associations) by using standard measurement methods such as "count" or "graph length". For the definition of derived measures and indicators the tool includes an evaluator of arithmetical and logical expressions, as Fig. 4 shows.

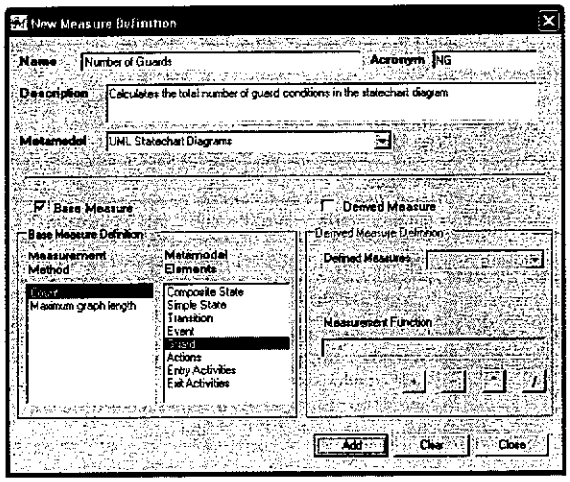


Fig. 4. New measure definition in GenMETRIC

6 Conclusions

In this work, we have illustrated the relevance of defining valid measures in a methodological way, following three main steps:

- **Measure definition.** We defined the measures with the idea of finding indicators of the understandability of UML statechart diagrams. Firstly, we defined them in natural language. Later, we have formally defined the measures using OCL and Maude, in order to alleviate a set of problems well-known by the Software Engineering community.
- **Theoretical validation.** We have also theoretically validated the measures using two different approaches, one based on properties and another based on the Measurement Theory.
- **Empirical validation.** We have presented three different families of experiments that were carried out in order to empirically check the validity of the measures previously presented. The first family provided as a result that a group of the measures were highly correlated with the understandability efficiency of UML statechart diagrams, as well as a preliminary prediction model for the understandability efficiency. The second family studied the effect that composite states have on the understandability of UML statechart diagrams. The results obtained indicate that, quite surprisingly, these structures do not improve significantly the understandability of the diagrams, at least in the conditions that we have used in our experimentation process. Agreeing with the previous family, the third family has allowed us conclude that using a flat nesting level within composite states makes an UML statechart diagram more understandable.

Finally, we have introduced GenMETRIC, a generic and extensible tool for the definition and automatic calculation of the different measures.

As future work, we are aware that it is necessary to continue refining the proposed measures and even to define some new ones (if necessary). This way we could have an adequate and useful set of measures for measuring the quality characteristics of UML statechart diagrams. Further empirical studies must also be preformed in order to reach a definitive and strong empirical validation of all the measures.

We will also try to extend the definition of measures to some other quality characteristics, such as modifiability, that also affect the maintainability of the diagrams.

Acknowledgements

This research is part of the MECENAS project (PBI06-0024) financed by "Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha" and the ESFINGE project (TIN2006-15175-C05-05) and the CALIPSO Network (TIN20005-24055-E) supported by the "Ministerio de Educación y Ciencia" (Spain).

References

1. Baroni, A.L., Braz, S., and Brito e Abreu, F. Using OCL to Formalize Object-Oriented Design Metrics Definitions. In *Proceedings of 6th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2002)* (Malaga, Spain, 2002). 99-106.
2. Briand, L., El-Emam, K., and Morasca, S., *Theoretical and Empirical Validation of Software Product Measures*. 1995, ISERN.
3. Briand, L., Morasca, S., and Basili, V. Property-Based Software Engineering Measurement. *IEEE Transactions on Software Engineering, 22,* 1 (1996), 68-86.
4. Briand, L., Morasca, S., and Basili, V. Response to: Comments on "Property-Based Software Engineering Measurement: Refining the Additivity Properties". *IEEE Transactions on Software Engineering, 23,* (1997), 196-197.
5. Briand, L., Wüst, J., Daly, J., and Porter, V. Exploring the Relationships between Design Measures and Software Quality in Object-Oriented Systems. *The Journal of Systems and Software, 51,* (2000), 245-273.
6. Briand, L., Wüst, J., and Lounis, H., *Investigating Quality Factors in Object-oriented Designs: An Industrial Case Study*. 1998, Technical Report ISERN 98-29, version 2.
7. Calero, C., Piattini, M., and Genero, M. Method for Obtaining Correct Metrics. In *Proceedings of 3rd International Conference on Enterprise and Information Systems (ICEIS 2001)* (Setúbal, Portugal, 2001). 779-784.
8. Cartwright, M. An Empirical View of Inheritance. *Information and Software Technology, 40,* 4 (1998), 795-799.
9. Clavel, M., Durán, F., Eker, S., Lincoln, P., Marínez-Oliet, N., Meseguer, J., and Talcott, C., *Maude 2.1.1 Manual*. 2003, University of Illinois at Urbana-Champaign.
10. Cruz-Lemus, J.A., Genero, M., and Piattini, M. Investigating the Nesting Level of Composite States in UML Statechart Diagrams. In *Proceedings of 9th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE'05)* (Glasgow, United Kingdom, 2005). 97-108.
11. Cruz-Lemus, J.A., Genero, M., Piattini, M., and Morasca, S. Improving the Experimentation for Evaluating the Effect of Composite States on the Understandability of UML State-

chart Diagrams. In *Proceedings of 5th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2006)* (Rio de Janeiro, Brazil, 2006). 9-11.
12. Cruz-Lemus, J.A., Genero, M., Piattini, M., and Toval Álvarez, J.A. An Empirical Study of the Nesting Level of Composite States within UML Statechart Diagrams. In *Proceedings of First International Workshop on Best Practices of UML (BP-UML 2005) - ER 2005 Workshops* (Klagenfurt (Austria), 2005). 12-22.
13. Cruz-Lemus, J.A., Maes, A., Genero, M., Poels, G., and Piattini, M., *The Impact of Structural Complexity on the Understandability of UML Statechart Diagrams*. 2007, University of Ghent.
14. Daly, J., Brooks, A., Miller, J., Roper, M., and Wood, M. An Empirical Study Evaluating Depth of Inheritance on Maintainability of Object-Oriented Software. *Empirical Software Engineering, 1*, 2 (1996), 109-132.
15. García, F., Serrano, M., Cruz-Lemus, J.A., Ruiz, F., and Piattini, M. Managing Software Process Measurement: a Metamodel-based Approach. *Information Sciences, 177*, (2007), 2570-2586.
16. Mayer, R.E., *Multimedia Learning*, Cambridge University Press, 2001.
17. OMG, *UML 2.0 OCL Final Adopted Specification*. 2005, Object Management Group.
18. Poels, G. and Dedene, G., *Distance: A Framework for Software Measure Construction*. 1999, Department of Applied Economics, Catholic University of Leuven, Belgium.
19. Poels, G. and Dedene, G. Evaluating the Effect of Inheritance on the Modifiability of Object-Oriented Business Domain Models. In *Proceedings of 5th European Conference on Software Maintenance and Reengineering (CSMR 2001)* (Lisbon, Portugal, 2001). 20-29.
20. Prechelt, L., Unger, B., Philippsen, M., and Tichy, W. A Controlled Experiment on Inheritance Depth as a Cost Factor for Code Maintenance. *The Journal of Systems and Software, 65*, (2003), 115-126.
21. Webb, K., *Xholon Digital Watch Project*. 2006.