Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Manfred A. Jeusfeld Óscar Pastor (Eds.)

Conceptual Modeling for Novel Application Domains

ER 2003 Workshops ECOMO, IWCMQ, AOIS, and XSDM
Chicago, IL, USA, October 13, 2003
Proceedings

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Manfred A. Jeusfeld
Tilburg University
Department of Information Systems and Management
P.O. Box 90153, Tilburg, 5000 LE, The Netherlands
E-mail: jeusfeld@uvt.nl

Óscar Pastor
Polytechnical University of Valencia
Camino de Vera s/n, Valencia, 46022 Spain
E-mail: opastor@dsic.upv.es

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

CR Subject Classification (1998): H.2, H.3, H.4, H.5, K.4.4, I.2

ISSN 0302-9743
ISBN 3-540-20257-9 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by PTP Berlin GmbH
Printed on acid-free paper SPIN: 10950012 06/3142 5 4 3 2 1 0

Preface

ER 2003, the 22nd International Conference on Conceptual Modeling in Chicago, Illinois, hosted four workshops on emerging and maturing aspects of conceptual modeling. While the entity-relationship approach is used to address data (base) modeling, the increasingly connected information infrastructure demands answers that can handle complexity and can develop models about systems that are maintainable. We received seven excellent proposals for workshops to be held at ER 2003, out of which we selected the following four based on peer reviews:

- Conceptual Modeling Approaches for E-Business (eCOMO 2003) brought together researchers and practitioners interested in conceptual modeling techniques for e-business.
- The International Workshop on Conceptual Modeling Quality (IWCMQ 2003) concentrated on approaches to quality assurance in the modeling process.
- The International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS 2003) was devoted to investigating the agent paradigm for information systems development.
- Finally, the International Workshop on XML Schema and Data Management (XSDM 2003) addressed the impact of XML on topics like data integration, change management, and the Semantic Web.

All four workshops highlighted relatively new viewpoints on conceptual modeling. Conceptual modeling as such has been greatly influenced and shaped by the entity-relationship model of Peter Chen. However, new developments like object-orientation and the World-Wide Web require adaptions and new techniques. No longer can developers assume that they can completely understand or model the information system. The new developments create challenges in various directions; some of these were discussed in detail in the four ER 2003 workshops:

E-Business and E-Commerce. The rise of the Internet has created new opportunities for defining and enacting business relations between partners. The question is how information systems can help in finding business partners, creating new services, and enacting those new services. Any lack of information about some business partners or their products and services needs to be compensated for using some kind of trust-building institution or mechanism. Moreover, services for e-business are not necessarily linked tightly togethers, as used to be the case for information systems developed for single enterprises. Can a service be modeled independently from the provider of the service who is selected at run time? Last but not least, one has to take into account different business (process) models, business contracts, and their monitoring. Hence, the field of e-business stresses the need for comprehensive modeling and analysis techniques.

Model Quality. Conceptual models are products of modeling processes undertaken by a group of human experts. Industrial quality management has shifted from quality tests at the end of the production process to quality assurance over all product development steps, including the early stages of requirements analysis. The same idea is being applied to improving or at least assessing the quality of conceptual models and the related modeling processes that create them. The more that such models are abstracted from the final implementation, the more difficult it appears to be to assess and control their quality. What constitutes an error in a model? Can we distinguish useful parts of a conceptual model from not so useful parts? Certainly, a team of modelers who are aware of the quality of their products has better opportunities to improve than a team of modelers who are not assessing quality aspects at all. Still, the questions are: what aspects to measure, with which methods, and how frequently?

Agent Orientation. Object-orientation is a programming and modeling paradigm that aims at encapsulation (hiding internal details) and re-use (of code and models). While this paradigm is still successful and valid, the lack of information about some components of an information system makes it less applicable to loosely coupled system, like Web services or complex factories that are under constant evolution. Agent orientation provides a promising approach to deal with the increased complexity by including a flavor of autonomy into the components of an agent-oriented system: the co-operating agents have goals and they govern over multiple possible strategies to achieve their goals. The challenge from a conceptual modeling perspective is to represent agent systems in a way that makes them subject to analysis. Suitable languages from agent communication, goal representation, etc., are still under development.

XML Data and Schema. The last, but not least, topic covered by the ER 2003 workshops is XML. XML was, after the revolutionary rise of the Internet, in particular the World-Wide Web, an attempt to bring some order into the Web by tagging data elements with labels that indicate their interpretation (or schema). In a way, it is the global representation of interoperable data and perhaps processes. But does XML solve the problems of data/schema integration or does it just shift the problem to a new (yet uniform) syntax? XML databases are already on the market, including XML-based query languages. So, what parts of the traditional data modeling theory can be translated for the XML case?

The ER 2003 workshops addressed these issues and created a forum for fruitful discussions. The fact that three of the four workshops have already a long history shows that such discussions are long-term, and convincing answers will only appear after some time.

We thank our colleagues in the ER 2003 organization committee for their support. In particular, we thank the organizing chairs of the four workshops who came up with the ideas and imagination that made the workshop program at ER 2003 possible. Last but not least, our special thanks go to the paper authors and the reviewers who created the content of this volume and ensured its high quality.

October 2003

Manfred Jeusfeld
Óscar Pastor

ER 2003 Workshop Organization

General

ER 2003 Workshops Chairs	Manfred A. Jeusfeld Tilburg University, The Netherlands Óscar Pastor Politechnical University of Valencia, Spain

eCOMO 2003 Organization

Heinrich C. Mayr	University of Klagenfurt, Austria
Willem-Jan van den Heuvel	Tilburg University, The Netherlands
Christian Kop	University of Klagenfurt, Austria

IWCMQ 2003 Organization

Jim Nelson	Ohio State University, USA
Geert Poels	Ghent University, Belgium
Marcela Genero	Universidad de Castilla, Spain
Mario Piattini	Universidad de Castilla, Spain

AOIS 2003 Organization

Paolo Giorgini	University of Trento, Italy
Brian Henderson-Sellers	University of Technology, Sydney, Australia

XSDM 2003 Organization

Sanjay Madria	University of Missouri-Rolla, USA

eCOMO 2003 Program Committee

Fahim Akhter	Zayed University, United Arab Emirates
Boldur Barbat	Lucian Blaga University, Sibiu, Romania
Boualem Benatallah	University of New South Wales, Sydney, Australia
Anthony Bloesch	Microsoft Corporation, USA
Antonio di Leva	University of Torino, Italy
Vadim A. Ermolayev	Zaporozhye State University, Ukraine
Marcela Genero	University of Castilla-La Mancha, Ciudad Real, Spain
Martin Glinz	University of Zurich, Switzerland
József Györkös	University of Maribor, Slovenia
Bill Karakostas	City University, London, UK
Roland Kaschek	Massey University, New Zealand
Stephen Liddle	Brigham Young University, USA
Zakaria Maamar	Zayed University, United Arab Emirates
Norbert Mikula	Intel Labs, Hillsboro, USA
Óscar Pastor	University of Valencia, Spain
Barbara Pernici	Politecnico di Milano, Italy
Matti Rossi	Helsinki School of Economics, Finland
Michael Schrefl	University of Linz, Austria
Daniel Schwabe	PUC-Rio, Brazil
Il-Yeol Song	Drexel University, Philadelphia, USA
Bernhard Thalheim	BTU, Cottbus, Germany
Jos van Hillegersberg	Erasmus University, Rotterdam, The Netherlands
Ron Weber	University of Queensland, Australia
Carson Woo	UBC, Vancouver, Canada
Jian Yang	Tilburg University, The Netherlands

IWCMQ 2003 Program Committee

Deb Armstrong	University of Arkansas, USA
Sjaak Brinkkemper	Baan, The Netherlands
Giovanni Cantone	University of Rome, Italy
Guido Dedene	Katholieke Universiteit Leuven, Belgium
Brian Henderson-Sellers	University of Technology, Sydney, Australia
Paul Johannesson	Stockholm University, Sweden
Barbara Kitchenham	Keele University, UK
John Krogstie	Sintef, Norway
Heinrich Mayr	University of Klagenfurt, Austria
Daniel Moody	Norwegian University of Science and Technology, Norway
Jim Nelson	Ohio State University, USA
Jeff Parsons	Memorial University of Newfoundland, Canada
Óscar Pastor	University of Valencia, Spain
Gustavo Rossi	National University of La Plata, Argentinia
Houari Sahraoui	Université de Montreal, Canada
Reinhard Schuette	University of Essen, Germany
Keng Siau	University of Nebraska-Lincoln, USA
Guttorm Sindre	Norwegian University of Science and Technology, Trondheim, Norway
Monique Snoeck	Katholieke Universiteit Leuven, Belgium
Bernhard Thalheim	Brandenburg University of Technology at Cottbus, Germany

Table of Contents

Conceptual Modeling Approaches for E-Business at ER 2003 (eCOMO 2003)

Second International Workshop on Conceptual Modeling Quality at ER 2003 (IWCMQ 2003)

Agent-Oriented Information Systems at ER 2003 (AOIS 2003)

International Workshop on XSDM at ER 2003 (XSDM 2003)

Acknowledgements. The VITI project was founded by CICYT and several other enterprises.

References

1. Bernus, P.: Business Evolution and Enterprise Integration. In: Kosanke, K., Nell, J. (eds.): Proceedings ICEIMT97. Springer Verlag, Berlin (1997) 140–151
2. Hawa, M., Ortiz, A., Lario, F., Ros, L.: Improving the role played by humans in the development of enterprise engineering and integration projects. International Journal of Computer Integrated Manufacturing. 15–4 (2002) 335–344
3. Camarinha-Matos, L.M., Afsarmanesh, H.: Elements of a base VE infrastructure. Computers in Industry. (Accepted to be published in 2003)
4. Yoo, S.B., Kim, Y.: Web-based knowledge management for sharing product data in virtual enterprises. International Journal of Production Economics. 75 (2002) 173–183
5. AMICE: CIMOSA: Open System Architecture for CIM. ESPRIT Consortium AMICE. Springer-Verlag, Berlin (1993)
6. Doumeingts, G., Vallespir, B., Zanettin, M., Chen, D.: GIM, GRAI Integrated Methodology, A Methodology for Designing CIM Systems. A technical report of the IFAC/IFIP Task Force on Architectures For Integrating Manufacturing Activities And Enterprises, GRAI/LAP, Version 1.0, Bordeaux (1992)
7. Williams, T.: The Purdue Enterprise Reference Architecture. Proceedings of the Workshop on Design of Information Infrastructure Systems for manufacturing. Elsevier Science, Tokyo (1993)
8. Poler, R., Lario, F.C.: Dynamic Model of Decision Systems (DMDS). Computers in Industry. 49 (2002) 175–193
9. Ortiz, A., Lario, F., Ros, L.: Enterprise Integration-Business Processes Integrated Management: a proposal for a methodology to develop Enterprise Integration Programs. Computers in Industry. 40 (1999) 155–171
10. Ortiz, A., Lario, F., Ros, L., Hawa, M.: Building a production planning process using an approach based on CIMOSA an workflow management systems. Computers in Industry. 40 (1999) 207–219
11. Chalmeta, R., Grangel, R.: ARDIN Extension for Virtual Enterprise Integration. Journal of Systems and Software. (Accepted to be published in 2003)

Second International Workshop on Conceptual Modeling Quality (IWCMQ 2003)

at the 22nd International Conference on Conceptual Modeling (ER 2003)

Chicago, Illinois, October 13, 2003

Organized by

Jim Nelson, Geert Poels, Marcela Genero, and Mario Piattini

Preface to IWCMQ 2003

This section of the ER'03 workshop proceedings includes the papers accepted for the Second International Workshop on Conceptual Modeling Quality (IWCMQ'03) held in Chicago, Illinois, USA, on October 16, 2003.

Conceptual modeling has been recognized as a key task that lays the foundation of all later design and implementation work. The early focus on conceptual modeling may help building better systems, without unnecessary rework at later stages of the development when changes are more expensive and more difficult to perform. Quality in conceptual modeling has been a topic of research since the early nineties but recently a stronger emphasis is given to the assessment, evaluation, and improvement of the models produced in the early phases of the system development life cycle.

IWCMQ'03 invited papers that explored the foundations of conceptual modeling quality, and methods for assessing, evaluating, and improving conceptual modeling quality. The workshop provided a forum for researchers and practitioners working in these and other areas to meet to discuss and push the boundaries of this very important and active area of research. We received a total of sixteen papers which were reviewed each by at least three program committee members. After a rigorous review process, seven papers were accepted. These papers, which are included in this section, fall into two broad categories.

The first category explores theoretical aspects of conceptual modeling quality with explorations into web conceptual design models (Franca Garzotto and Vito Perrone of the Politecnico di Milano, Italy), reference models (Peter Fettke and Peter Loos of the Johannes Gutenberg-University Mainz, Germany), UML behavioral diagrams (Marcela Genero, David Miranda, and Mario Piattini of the University of Castilla-La Mancha, Spain), and multidimensional schemas (Samira Si-Said Cherfi of CEDRIC-CNAM and Nicolas Prat of ESSEC, France).

The second describes conceptual model quality in practice in the fields of model consistency checking (Monique Snoeck, Cindy Michiels, and Guido Dedene of the Katholieke Universiteit Leuven, Belgium), query formulation (Hannu Jaakkola of Tampere University of Technology, Finland, and Bernhard Thalheim of the Brandenburg University of Technology at Cottbus, Germany), and the modeling of accounting systems (Geert Poels of Ghent University, Belgium).

We would like to express our thanks to the program committee members for their rigorous reviews of the papers, the ER'03 organizing committee -especially the Conference Chair and the Workshop Chairs- for their help and support, all the authors who submitted papers, the invited speaker, and the attendees

Defining Metrics for UML Statechart Diagrams in a Methodological Way

Marcela Genero, David Miranda, and Mario Piattini

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

Abstract. The fact that the usage of metrics at early phases of OO development can help designers make better decisions is gaining relevance. Moreover, the necessity of having early indicators of external quality attributes, such as understandability, based on early metrics is growing. There exists several works related to metrics for UML structural diagrams such as class diagrams. However, UML behavioral diagrams metrics have been disregarded in the software measurement arena. This fact leaded us to define a set of metrics for the size and structural complexity of UML statechart diagrams. Apart from the definition of the metrics, a contribution of this study is the methodological approach that was followed to theoretically validate them and to empirically validate them as understandability indicators.

Keywords: OO Software, UML statechart diagrams, understandability, maintainability, structural complexity, size, metrics, theoretical validation, empirical validation, experiment replication

1 Introduction

It is widely recognised that structural properties of OO software artefacts obtained at early phases of the development has a great influence on the quality of the product that is finally implemented. For this reason several proposals of metrics exists that can be applied to measure the size, structural complexity, coupling, etc. of UML class diagrams [11, 15, 20, 27, 28] and use case diagrams [23, 28].

However, there is little reference to metrics for behavioural diagrams such as UML statechart diagrams in the existing literature. One of the first approaches towards the definition of metrics for behavioural diagrams can be found in [17], where metrics were applied to statechart diagrams developed with OMT [34]. Yacoub et al. [40] proposed structural complexity and coupling metrics for measuring the quality of dynamic executions. These metrics were defined basing in concepts as Petri Net and McCabe's cyclomatic structural complexity and were applied to simulated scenarios in Real-Time Object Modelling (ROOM) [35]. Poels and Dedene [33] defined

empirical validation (except Poels and Dedene who performed the theoretical validation). The lack of metrics for diagrams that capture dynamics aspects of OO software motivated us to define metrics for behavioural diagrams, starting with metrics for UML statechart diagrams [21].

The aim of this paper is to define a set of metrics for measuring the size and structural complexity of UML statechart diagrams and investigate through experimentation if they are related the understandability of UML statechart diagrams[1]. If such a relationship exists and is confirmed by empirical studies, we will have really obtained early indicators of UML statechart diagram understandability. We consider the understandability because it is an external quality attribute which directly influence several quality characteristics [24], among others maintainability.[2]

The definition and the theoretical and empirical validation of the metrics have been done in a disciplined manner following a method that emerged as a combination of two proposals [13, 14].

The rest of this paper is organised in the following way: Section 2 presents the identification of metric goals and the proposal of metrics for UML statechart diagrams. Section 3 and 4 present the theoretical and the empirical validation of the proposed metrics, respectively. The paper ends with some conclusions and outlines the direction of our future research work.

2 Metric Definition

Using the GQM [1, 2] template for goal definition, the goal pursued for the definition of the metrics for UML statechart diagrams is:

> Analyse UML *statechart diagrams* for the purpose of *Assessing* with respect to their *Maintainability* from the point of view of the *Conceptual modellers*, OO software designers in the context of *Software development organisations*.

Following this goal we have defined a set of metrics each one focusing on a different UML statechart diagram elements [21] (see table 1).

3 Theoretical Validation of the Proposed Metrics

For the theoretical validation of the proposed metrics we followed Briand et al.'s framework [4] as a property-based framework, and Poels and Dedene's framework [32] as measurement theory-based framework.

Briand et al.'s framework [4] provides a set of mathematical properties that characterise and formalise several important measurement concepts such as size, length, complexity, cohesion and coupling, related to internal software attributes.

---

[1] The theoretical basis for developing quantitative models relating structural properties (size and structural complexity) and external quality ...

Table 1. Metrics for UML statechart diagrams

	Metric Name	Metric definition
Size metrics	**NEntryA** (Number of entry actions)	The total number of entry actions, i.e. the actions performed each time a state is entered
	NexitA (Number of exit actions)	The total number of exit actions, i.e. the actions performed each time a state is left
	NA (Number of activities)	The total number of activities (do/activity).
	NSS (Number of simple states)	The total number of states considering also the simple states within the composites states
	NCS (Number of composite states)	The total number of composite states.
	NE (Number of events)	The total number of events.
	NG (Number of guards)	The total numbers of guard conditions.
Structural complexity metrics	**McCabe** (Cyclomatic Number of McCabe [29][3].	It is defined as |NSS|-|NT|+2
	NT (Number of transitions)	The total number of transitions, considering common transitions (the source and the target states are different), the initial and final transitions, selftransitions (the source and the target states are the same) and internal transitions (transitions inside a state that responds to an event but without leaving the state).

Theoretical validation of NSS Metric. For our purpose and in accordance with Briand et al.'s framework [4], we consider that a statechart diagram is a system composed of states (elements) and transitions (relations). A module is composed of a subset of the states and transitions. We will demonstrate that NSS fulfils all of the axioms that characterise size metrics, as follows:

- *Nonnegativity.* The number of states in an statechart diagram is always greater than zero, so that NSS can never be negative.
- *Null value.* If we have no states NSS=0.
- *Module additivity.* If we consider that a statechart diagram is composed of modules with no states in common, the number of states of an statechart diagram it will be always the sum of the number of states of its modules.

Following an analogous reasoning used for NSS metric, it can be proved that the other metrics related to internal transitions, such as NCS and NE are also size metrics.

Theoretical validation of NA Metric. We consider that states are system modules, the activities are the elements and relationships are represented by the relation

[3] Even tough the Cyclomatic Number of McCabe was defined to calculate single module

"belong to", which reflects that each activity belongs to a state. We will demonstrate that NA fulfils all of the axioms that characterise size metrics, as follows:

- *Nonnegativity.* One state can or cannot have activities, i.e. it could happen that NA=0 or NA>0, but never NA<0.
- *Null value.* If we have no activities then NA=0.
- *Module additivity.* If we consider that a state is composed of substates (modules) with no activities in common, the NA of a state will always be the sum of the NA of all its substates, because each activity of a substate is an activity of the state.

In a similar way it can be proved that the metrics NEntryA, NexitA and NG are also size metrics.

Theoretical validation of NT metric. We consider that a statechart diagram is a system composed of states (elements) and transitions (relations). A module is composed of a subset of the statechart diagram states and statechart diagram transitions. We will demonstrate that NT fulfils all of the axioms that characterise complexity metrics, as follows:

- *Nonnegativity.* It is obvious that there is always a null or positive value of transitions. Then, NT≥0.
- *Null value.* If there are no transitions within an statechart diagram then NT = 0.
- *Symmetry.* The number of transitions does not depend on the convention used to represent the transitions.
- *Module monotonicity.* According to the definition of this property, it is obvious that: being m_1 and m_2 any two modules of the statechart diagram with no transitions in common, have the value of $NT(SD) \geq NT(m_1) + NT(m_2)$.
- *Disjoint module additivity.* Let m_1 and m_2 be any two disjoint modules such that $SD=m_1 \cup m_2$. Let NT_1 and NT_2 be the number of transitions in the m_1 and m_2 modules.

Obviously: $NT=NT_1+NT_2$, because m_1 and m_2 are disjoint modules. A property- based approach such as Briand et al.'s framework propose a measure property set that is necessary but not sufficient [4, 32]. They can be used as a filter to reject proposed measures [25], but they are not sufficient to prove the validity of the measure.

A measurement theory-based approach to software metric validation, like DISTANCE [32] offers a measure construction procedure to model properties of software artefacts and define the corresponding software metrics. An important pragmatic consequence of the explicit link with measurement theory is that the resulting measures define ratio scales.

Due to space constraints we cannot present the measurement construction process for the proposed metrics. A detailed validation can be found in [21].

4 Empirical Validation of the Proposed Metrics

In this section we describe an experiment and its replication, we carried out to empirically validate the proposed metrics as early understandability indicators.

We have followed some suggestions provided in [7, 26, 31, 39] on how to perform controlled experiments and have used (with only minor changes) the format proposed

Definition. Using the GQM template for goal definition, the goal of the experiment is the following:

> Analyse *UML statechart diagrams structural complexity and size metrics* for the purpose *Evaluating* with respect to their *capability of being used as understandability indicators of UML statechart diagrams* from the point of view of the *OO software modellers and OO software designers* in the context of *Undergraduate students in the final year of Computer Science and teachers of the Area of Software Engineering at the Department of Computer Science in the University of Castilla-La Mancha*

Planning. The planning includes the following activities:

- **Context selection.** The subjects were eight teachers and eleven students. Students are enrolled in the final-year of Computer Science at the Department of Computer Science in the University of Castilla-La Mancha in Spain. All of the teachers belong to the Software Engineering area. The experiment is specific since it focuses on UML statechart diagram structural complexity and size metrics. The ability to generalise from this specific context is further elaborated below when we discuss threats to the experiment. The experiment addresses a real problem, i.e., which indicators can be used to assess the understandability of UML statechart diagram? To this end it investigates the correlation between metrics and understandability.
- **Selection of subjects.** The subjects are chosen for convenience, i.e. the subjects are students that have medium experience in the design and development of OO software.
- **Variables selection.** The independent variables are UML statechart diagram structural complexity and size and the dependent variable is UML statechart diagram understandability.
- **Instrumentation.** The objects used in the experiment were 20 UML statechart diagrams. The independent variable was measured by the metrics presented in section 2. The dependent variable was measured by the time the subject spent answering the questionnaire attached to each diagram. We called that time "understandability time".
- **Hypothesis formulation**. An important aspect of experiments is to know and to state in a clear and formal way what we intend to evaluate in the experiment. This leads us to the formulation of the following hypotheses:
  - *Null hypothesis, H_0*: There is no significant correlation between the UML statechart diagrams structural complexity and size me trics and the understandability time.
  - *Alternative hypothesis, H_1*: There is a significant correlation between the UML statechart diagrams structural complexity and size metrics and the understandability time.
- **Experiment design.** We selected a within-subject design experiment, i.e. all the questionnaires had to be solved by each of the subjects. The subjects were given the tests in different order.

Operation. It is in this phase where measurements are collected including the following activities:

- **Preparation.** At the time the experiment was done all of the students had taken a software using UML. Moreover, the subjects were given an intensive training session before the experiment took placing. However, the subjects were not aware of what aspects we intended to study. Neither they were informed about the actual hypotheses stated.

  We prepared the material we handed to the subjects, which consisted of a guide explaining the UML statechart notation and 20 UML statechart diagrams. These diagrams were related to different universes of discourse that were easy enough to be understood by each of the subjects. The structural complexity and size of each diagram is different, covering a broad range of the metrics values.

  Each diagram had a test enclosed, which includes a questionnaire in order to evaluate if the subjects really understand the content of the UML statechart diagrams. Each questionnaire contained exactly the same number of questions (four) and the questions were conceptually similar and were written in identical order. Each subject had to write down the time he started answering the questionnaire and at the time they finished. The difference between the two is what we called the understandability time (expressed in seconds).
- **Execution.** The subjects were given all of the material described in the previous paragraph. We explained to them how to carry out the experiment. We allowed them one week to do the experiment, i.e., each subject had to carry out the test alone, and could use unlimited time to solve it. We collected all of the data with the understandability time calculated from the responses of the experiments.
- **Data Validation.** Once the data were collected, we controlled if the tests were completed and if the questions have been answered correctly. All the tests were considered valid because all the questions were correctly answered.

Analysis and Interpretation. First we summarized the data collected for each diagram. We had the metric values and we calculated the mean of the subjects' understandability time for each statechart diagram.[4]

First, we applied the Kolmogrov-Smirnov test to ascertain if the distribution of the data collected was normal. As the data were non-normal we decided to use a nonparametric test like Spearman's correlation coefficient, with a level of significance $\alpha = 0.05$, which means the level of confidence is 95% (i.e. the probability that we reject H_0 when H_0 is false is at least 95%, which is statistically acceptable). Each of the metrics was correlated separately to the mean of the subjects' understandability time (see table 2).

Table 2. Spearman's correlation coefficients between metrics and understandability time

NEntryA	NExitA	NA	NSS	NCS	NT	NE	NG	McCabe
0.1808	-0.2521	**0.4830**	**0.4999**	0.3352	**0.6049**	0.4261	**0.5535**	0.0773

For a sample size of 20 (mean values for each diagram) and $\alpha = 0.05$, the Spearman cut-off for accepting H_0 is 0.44 [5, 16]. Because the computed Spearman's correlation coefficients for metrics NA, NS and NT (see table 2), are above this cutoff and the p-value < 0.05, the null hypothesis H_0, is rejected.

Given these results, we can conclude that there is a significant correlation between NA, NSS, NT and NG metrics and subjects' understandability time.

Part of the information that these metrics provide might be redundant, which in statistical terms is equivalent to saying that metrics might be very correlated. This justifies the interest of analyzing the information that each metric captures to eliminate such redundancy. In the experimental research in software engineering [6, 9] problem is solved by using the Principal Component Analysis (PCA) [18] . In this case, through the PCA, the purpose is to reduce the space of 11 metric dimensions that contain the initial information.

Form the results of the PCA we may conclude that the rotated principal components are difficult to interpret, and it is too premature to decide if which of the metrics we proposed are redundant. Fact that confirm what is already known [6, 9], that the results obtained in the PCA are dependent on the data. So further investigation is needed too obtain stronger findings and decide if some of the metrics are redundant or not.

Validity evaluation. We will discuss the various issues that threaten the validity of the empirical study and how we attempted to alleviate them:

- Threats to Conclusion Validity. The conclusion validity defines the extent to which conclusions are statistically valid. The only issue that could affect the statistical validity of this study is the size of the sample data (20 values), which perhaps are not enough for both parametric and non-parametric statistic test [5].
  We are aware of this, so we will try to obtain bigger sample data through more experimentation.
- Threats to Construct Validity. The construct validity is the degree to which the independent and the dependent variables are accurately measured by the measurement instruments used in the experiment. For the dependent variable we use the understandability time, i.e., the time each subject spent answering the questions related to each diagram, that it is considered the time they need to understand it. It is an objective measure so we consider the understandability time could be considered a measure constructively valid. The construct validity of the metrics used for the independent variable is guaranteed by the theoretical validation we carried out with them (see section 3).
- Threats to Internal Validity. The internal validity is the degree of confidence in a cause-effect relationship between factors of interest and the observed results. The analysis performed here is correlational in nature. We have demonstrated that several of the metrics investigated had a statistically and practically significant relationship with understandability. Such statistical relationship do not demonstrate per se a causal relationship. They only provide empirical evidence of it. Only controlled experiments, where the metrics would be varied in a controlled manner and all other factors would be held constant, could really demonstrate causality. However, such a controlled experiment would be difficult to run since varying structural complexity and size in a system, while preserving its functionality, is difficult in practice. On the other hand, it is difficult to imagine what could be alternative explanations for our results besides a relationship between structural complexity, size and understandability. The following issues have also been dealt with: Differences among subjects, Knowledge of the universe of discourse among class diagrams, Precision in the time values, Learning effects, Fatigue effects, Persistence effects, Subject motivation, Plagiarism Influence between students, etc.

- Threats to External Validit y. External validity is the degree to which the research results can be generalised to the population under study and other research settings. The greater the external validity, the more the results of an empirical study can be generalised to actual software engineering practice. Two threats to validity have been identified which limit the ability to apply any such generalisation:
  - Materials and tasks used. In the experiment we tried to use statechart diagrams and tasks which can be representative of real cases, but more empirical studies taking "real cases" from software companies must be done in the future.
  - Subjects. To solve the difficulty of obtaining professional subjects, we used teachers and students from advanced software engineering courses. We are aware that more experiments with practitioners and professionals must be carried out in order to be able to generalise these results. However, in this case, the tasks to be performed do not require high levels of industrial experience, so, experiments with students could be appropriate [3].

Presentation and Package. As the diffusion of experimental data is important for the external replication of the experiments [12] we have put all of the material of this experiment onto the website http://alarcos.inf -cr.uclm.es .

4.1 Replication of the Experiment

In order to corroborate the findings obtained in the experiment previously described we carried out an internal strict replication [3, 12] of it. The most important difference between the previous experiment we carried out and this replication are:

- The subjects were undergraduate third-year student of Computer Science, which have had only one course of Software Engineering, where they learnt how to design OO software using UML. This means that the experience of the subjects is lesser.
- The subjects have to solve the tests alone, in no more than two hours. Any doubt, could be solved by the person who monitored the experiment. In fact, the replication was carried out in a more controlled environment due to the fact that it was supervised. Fact that can contribute to control the plagiarism between subjects.

After performing a correlational analysis using the data obtained in the replication we obtained the results shown in table 3.

Table 3. Spearman's correlation coefficients between metrics and understandability time (replication)

NEntryA	NExitA	NA	NSS	NCS	NT	NE	NG	McCABE
-0.04581	-0.34611	**0.51714**	**0.57474**	0.42809	**0.54980**	0.36412	**0.63063**	-0.03260

Comparing the findings of both experiments (see tables 2 and 3) we realized that they are similar. This means that the metrics NA, NSS, NG and NT are to some extent correlated with the understandability time of UML statechart diagrams.

5 Conclusions and Future Work

With the hypothesis that the size and the structural complexity of UML statechart diagrams may influence their understandability (and therefore in their maintainability), we defined a set of metrics for the structural complexity and size of UML statechart diagrams in a methodological way [21].

The theoretical validity of the proposed metrics, which means that they really measure the attribute they purport to measure was demonstrated trough the validation following two approaches: a property-based appro ach such as the the Briand et al.'s framework [4] and a measurement theory-based approach such as the DISTANCE framework [32]. Moreover, the use of DISTANCE guarantees that the metrics can be used as ratio scale measurement instruments.

Our hypothesis was to some extent empirically corroborated by a controlled experiment we carried out and its replication. As a result of all the experimental work, we can conclude that the metrics NA, NS, NG and NT seem to be highly correlated with the understandability of UML statechart diagrams.

Nevertheless, despite the encouraging results obtained we still consider them as preliminaries. Further replication, both internal and external, is of course necessary and also new experiments must be carried out with practitioners who work in software development organizations. Only after performing a family of experiments we can build an adequate body of knowledge to extract useful measurement conclusions regarding the use these metrics to be applied in real measurement projects as early understandability indicators of the UML statechart diagrams [3, 30, 36].

Once we obtained stronger results in this line, we think the metrics we proposed could also be used for allowing OO software modellers a quantitative comparison of design alternatives, and therefore, an objective selection among several statechart diagram alternatives with equivalent semantic content, and predicting external quality characteristic, like maintainability in the initial stages of the OO software life cycle and a better resource allocation based on these predictions. In this sense we plan to build a maintainability prediction model (based on the metrics values) using traditional statistical techniques and advances techniques borrowed from artificial intelligence.

Finally, another research line of interest would be to evaluate the influence of the structural complexity and size of the UML statechart diagrams on other maintainability factors such as modifiability and analysability.

Acknowledgements. This research is part of the DOLMEN (TIC 2000-1673-C06-06) and the CALDEA (TIC 2000-1673-C06-06) projects, financed by Subdirección General de Proyectos de Investigación, – Ministerio de Ciencia y Tecnología (in Spain).

References

1. Basili, V., Rombach, H.: The TAME project: towards improvement-oriented software environments. IEEE Transactions on Software Engineering 14(6) 1998 758–773

2. Basili, V., Weiss, D.: A Methodology for Collecting Valid Software Engineering Data. IEEE Transactions on Software Engineering 10(6) (1984) 728–738
3. Basili, V., Shull, F., Lanubille, F.: Building Knowledge through families of experiments. IEEE Transactions on Software Engineering 25(4) (1999) 456–473
4. Briand, L., Morasca, S., Basili, V.: Property-based software engineering measurement. IEEE Transactions on Software Engineering, 22 (1) (1996) 68–85
5. Briand, L., El Emam, K., Morasca, S.: Theoretical and empirical validation of software product measures. Technical Report ISERN-95-03, International Software Engineering Research Network (1995)
6. Briand, L., Wüst, J., Lounis, H.: Replicated Case Studies for Investigating Quality Factors in Object-oriented Designs. Technical report ISERN 98-29 (version 3), International Software Engineering Research Network (1998)
7. Briand, L., Arisholm, S., Counsell, F., Houdek, F., Thévenod-Fosse, P.: Empirical Studies of Object-Oriented Artifacts, Methods, and Processes: State of the Art and Future Directions. In Empirical Software Engineering 4(4) (1999) 387–404
8. Briand, L., Bunse, C., Daly, J.: Controlled Experiment for evaluating Quality Guidelines on the Maintainability of Object-Oriented Designs. IEEE Transactions on Software Engineering, 27(6) (2001) 513–530
9. Briand, L., Wüst, J.: Empirical studies of quality models.: Advances in Computers Academic Press, Zelkowitz (ed), 59 (2002) 97–166
10. Briand, L., Morasca, S., Basili, V.: An operational process for goal-driven definition of measures. IEEE Transactions on Software Engineering 28(12) (2002) 1106–1125
11. Brito e Abreu, F., Carapuça, R.: Object-Oriented Software Engineering: Measuring and controlling the development process., 4th Int Conference on Software Quality, Mc Lean, Va, USA, (1994)
12. Brooks, A., Daly, J., Miller, J., Roper, M., Wood, M.: Replication of experimental results in software engineering. In Technical report ISERN-96-10, International Software Engineering Research Network, (1996)
13. Calero, C., Piattini, M., Genero, M.: Method for obtaining correct metrics. International Conference on Enterprise and Information Systems (ICEIS`2001), (2001) 779–784
14. Cantone, G., Donzelli, P.: Production and maintenance of software measurement models. Journal of Software Engineering and Knowledge Engineering, 5 (2000) 605–626
15. Chidamber, S., Kemerer, C.: A Metrics Suite for Object Oriented Design. In IEEE Transactions on Software Engineering 20(6) (1994) 476–493
16. CUHK – Chinese University of Hong Kong – Department of Obstetrics and Gynaecology – http://department.obg.cuhk.edu.hk/ResearchSupport/Minimum_correlation.asp (Last visited on July 22nd, 2002).
17. Derr, K.: Applying OMT,SIGS Books, Prentice Hall. New York, (1995)
18. Dunteman, G.:Principal Component Analysis. Sage University Paper 07–69, Thousand Oaks, CA, (1989)
19. Fenton, N., Pfleeger, S.: Software Met rics: A Rigorous Approach, Chapman & Hall. London, 2nd. Edition, (1997)
20. Genero, M. Defining and Validating Metrics for Conceptual Model, Ph.D. thesis. University of Castilla–La Mancha, (2002)
21. Genero, M., Miranda, D., Piattini, M.: Defining and Validating Metrics for UML Statechart Diagrams. In 6th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE'2002), 120–136 (2002)
22. Harrison, R., Counsell, S., Nithi, R.: Experimental Assessment of the Effect of Inheritance on the Maintainability of Object-Oriented Systems. Journal of Systems and Software, 52 (2000) 173–179
23. Henderson-Sellers, B., Zowghi, D., Klemola, T., Parasuram, S. Sizing Use Cases: How to create a standard metrical approach. 8th International Conference (OOIS 2002), LNCS 2425 (2002) 409–421

24. ISO 9126: Software Product Evaluation-Quality Characteristics and Guidelines for their Use, ISO/IEC Standard 9126. Geneva (2001)
25. Kitchenham, B., Stell, J.: The danger of using axioms in software metrics. IEE Proc.-Soft. Eng. 144(5-6) (1997) 79–285
26. Kitchenham, B., Pflegger, S., Pickard, L., Jones, P., Hoaglin, D., El-Emam, K. and Rosenberg, J. Preliminary Guidelines for Empirical Research in Software Engineering. In IEEE Transactions of Software Engineering, 28(8) (2002) 721–734
27. Lorenz, M., Kidd, J. : Object-Oriented Software Metrics: A Practical Guide, Prentice Hall. Englewood Cliffs, New Jersey, (1994)
28. Marchesi, M.: OOA Metrics for the Unified Modeling Language. In 2nd Euromicro Conference on Software Maintenance and Reengineering, (1998) 67–73
29. McCabe, T.: A Complexity Measure. IEEE Transactions on Software Engineering, 2(4) (1976) 308–320
30. Miller, J.: Applying Meta-Analytical Procedures to Software Engineering Experiments. In Journal of Systems and Software, 54 (2000) 29–39
31. Perry, D., Porter, A., Votta, L.: Empirical Studies of Software Engineering: A Roadmap. In Future of Software Engineering. Anthony Finkelstein Ed., ACM, (2000) 345–355
32. Poels, G., Dedene, G. Distance-Based software measurement: necessary and sufficient properties for software measures. In Information and Software Technology, 42(1) (2000) 35–46
33. Poels, G., Dedene, G.: Measures for Assessing Dynamic Complexity Aspects of Object-Oriented Conceptual Schemes. In 19th International Conference on Conceptual Modelling (ER 2000), LNCS 1920, Springer-Verlag, (2000) 499–512.
34. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. , Lorensen, W.: Object-Oriented Modelling and Design, Prentice Hall. USA, (1991)
35. Selic, B., Gullekson, G., Ward, P.: Real-Time Object Oriented Modelling, John Wiley & Sons, Inc., (1994)
36. Shull, F., Basili, V., Carver, J., Maldonado, J.: Replicating Software Engineering Experiments: Addressing the Tacit Knowledge Problem. International Symposium on Empirical Software Engineering (ISESE 2002), Nara, Japan, IEEE Computer Society, (2002) 7–16
37. Snoeck, M.: On a process algebra approach for the construction and analysis of M.E.R.O.D.E.-based conceptual models, Ph.D. Katholieke Universiteit Leuven, (1995)
38. SPSS 11.0.: Syntax Reference Guide. Chicago. SPSS Inc., 2001
39. Wohlin, C., Runeson, P., Höst, M., Ohlson, M., Regnell, B.,Wesslén, A.: Experimentation in Software Engineering: An Introduction, Kluwer Academic Publishers, (2000)
40. Yacoub, S., Ammar, H., Robinson, T.: Dynamic Metrics for Object Oriented Designs. In Sixth IEEE International Symposium on Software Metrics, 1998

Visual SQL – High-Quality ER-Based Query Treatment

Hannu Jaakkola[1] and Bernhard Thalheim[2]

[1] Tampere University of Technology,
Pori, P.O.Box 300, FIN-28101 Pori,
hannu.jaakkola@tut.fi
[2] Computer Science Institute,
Brandenburg University of Technology at Cottbus,
PostBox 101344, D-03013 Cottbus,
thalheim@informatik.tu-cottbus.de

Abstract. Query formulation is still a difficult task whenever a database schema is large or complex. The user has to entirely understand the schema before a correct and complete formulation of the query will be found. Furthermore, users may overlook types in the SQL schema that must be used in the query. We show in this paper that visualization led in this case to higher conceptual correctness and conceptual completeness. Visualization is based on Visual SQL. Visual SQL follows the paradigm of entity-relationship representation. At the same time, it has the same expression power as SQL-92. The quality of query formulation is, however, higher.

1 Visualization Increases Correctness and Completeness of Queries

The improvement of database management systems and of computing power, the increase of the size of the databases under use, and the availability of distributed applications have led to new and more challenging applications. The size of database schemata has grown to a size that is far beyond an experienced programmer may survey. Therefore, new techniques are required to understand data structuring and to discover some knowledge from databases.

Visualization. Visualization of complex conceptual structures is a key component of support for many applications in science and engineering. An ER schema is an abstract structure that is used to model information. ER schemata are used to represent information that can be modeled as objects and connections between those objects.

Visualization of structures is only useful to the degree that the associated diagrams effectively convey information to the people that use them. A good