Lecture Notes in Computer Science

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

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

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

... sublines have been added featuring, ... added-value components; these ... include

- ... or collections of lectures given at ...
- ... complete and mediated coverage ...
- ... topics to the broader community)

... each new volume is published electronically ...

... can be found at ...

... should be sent to

... Heidelberg, Germany

ISSN 0302-9743

ISBN 978-3-540-87674-2

Lecture Notes in Computer Science

LNCS LNAI LNBI

Czarnecki et al. (Eds.)

LNCS 5301

... and Systems

MoDELS 2008

LNCS 5301

... Ileana Ober
... Axel Uhl

Model Driven Engineering Languages and Systems

11th International Conference, MoDELS 2008
Toulouse, France, September/October 2008
Proceedings

Springer

Lecture Notes in Computer Science 5301

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Krzysztof Czarnecki Ileana Ober
Jean-Michel Bruel Axel Uhl
Markus Völter (Eds.)

Model Driven Engineering Languages and Systems

11th International Conference, MoDELS 2008
Toulouse, France, September 28 - October 3, 2008
Proceedings

Springer

Volume Editors

Krzysztof Czarnecki
University of Waterloo
Department of Electrical and Computer Engineering
200 University Ave., West Waterloo, ON, N2L 3G1, Canada
E-mail: czarnecki@acm.org

Ileana Ober
Université Paul Sabatier, IRIT - MACAO
118, route de Narbonne, 31062 Toulouse, France
E-mail: ileana.ober@irit.fr

Jean-Michel Bruel
Université de Pau et des Pays de l'Adour
Département Informatique
Av. de l'Université, B.P. 1155, 64013 Pau, France
E-mail: Jean-Michel.Bruel@univ-pau.fr

Axel Uhl
SAP AG, 69190 Walldorf, Germany
E-mail: axel.uhl@sap.com

Markus Völter
Independent Consultant
Grabenstrasse 4, 73033 Göppingen, Germany
E-mail: voelter@acm.org

Library of Congress Control Number: 2008935624

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

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-540-87874-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-87874-2 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12534548 06/3180 5 4 3 2 1 0

Preface

MODELS 2008 was the 11th edition of the series of conferences on Model-Driven Engineering Languages and Systems. The conference was held in Toulouse, France, during the week of September 28 to October 3, 2008. The local arrangements were provided by the Institut de Recherche en Informatique de Toulouse (IRIT).

The conference program included three keynote presentations, technical paper presentations, two panels, and several workshops and tutorials. The invited keynote speakers were Don Batory (University of Texas, USA), Jeff Kramer (Imperial College London, UK), and Patrick Rauhut (Airbus, Germany).

This volume contains the final versions of the papers accepted for presentation at the conference. The papers cover a wide range of topics from the field including model transformation, model management, domain-specific modeling, modeling language semantics, model analysis, and applications.

We received a record number of 271 full paper submissions from 40 different countries. Of these, 43 papers were submitted by authors from more than one country. The top three countries submitting papers were France (40), Germany (38), and Canada (24). A total of 58 papers were accepted for inclusion in the proceedings. The acceptance rate was therefore 21%, which is somewhat lower than those of the previous MODELS conferences.

At least three Program Committee or Expert Reviewer Panel members reviewed each paper. Reviewing was thorough, and most authors received detailed comments on their submissions. Conflicts of interest were taken very seriously. No-one participated in any way in the decision process of any paper where a conflict of interest was identified. In particular, PC members who submitted papers did not have access to information concerning the reviews of their papers.

We would like to thank everyone who submitted papers as well as proposals for workshops and tutorials. We would also like to thank the large number of volunteers who contributed to the success of the conference. Richard van de Stadt deserves special thanks for his prompt and gracious service in supporting special requests for CyberChairPRO, the conference management system used to manage papers submissions and the virtual PC meeting. Finally, we would like to thank our sponsors, ACM and IEEE Computer Society, for their support of the MODELS 2008 conference.

October 2008

Krzysztof Czarnecki
Ileana Ober
Jean-Michel Bruel
Axel Uhl
Markus Völter

Organization

Conference Chairs

Ileana Ober (IRIT, France)
Jean-Michel Bruel (LIUPPA, France)

Program Chair

Krzysztof Czarnecki (University of Waterloo, Canada)

Experience Track Chairs

Axel Uhl (SAP, Germany)
Markus Völter (Independent Consultant, Germany)

Technological Track Chair

Pascal Roques (Valtech Training, France)

Workshop Chair

Michel Chaudron (Technical University Eindhoven and Leiden University, The Netherlands)

Tutorial Chair

Xavier Blanc (University Pierre et Marie Curie, France)

Panel Chair

Sudipto Ghosh (Colorado State University, USA)

Research Project Symposium Chair

Iulian Ober (Toulouse University, France)

Doctoral Symposium Chair

Alexander Pretschner (ETH Zurich, Switzerland)

Educators' Symposium Chair

Michał Śmiałek (Warsaw University of Technology, Poland)

Publicity Chair

Benoit Baudry (IRISA, France)

Web Chair

Nicolas Belloir (LIUPPA, France)

Organizing Committee

Jean-Paul Bodeveix (France)
Pierre Bazex (France)
Nicolas Belloir (France)
Agusti Canals (France)
Maura Cerioli (Italy)
Xavier Crégut (France)
Patrick Farail (France)
Louis Féraud (France)
Geri Georg (USA)
Hervé Leblanc (France)
Michel Lemoine (France)
Thierry Millan (France)
Mario Paludetto (France)
Christian Percebois (France)

Program Committee

João Araújo (Portugal)
Uwe Aßmann (Germany)
Benoit Baudry (France)
Xavier Blanc (France)
Jean Bézivin (France)
Paulo Borba (Brazil)
Lionel Briand (Norway)
Betty Cheng (USA)
Shigeru Chiba (Japan)
Krzysztof Czarnecki (Canada)
Juergen Dingel (Canada)
Gregor Engels (Germany)
Alexander Egyed (Austria)
Jean-Marie Favre (France)
Bernd Fischer (UK)
Robert France (USA)
Harald Gall (Switzerland)
Dragan Gašević (Canada)
Geri Georg (USA)
Holger Giese (Germany)
Tudor Girba (Switzerland)
Martin Gogolla (Germany)
Aniruddha Gokhale (USA)
Orla Greevy (Switzerland)
Paul Grünbacher (Austria)
John Grundy (New Zealand)
Øystein Haugen (Norway)
Simon Helsen (Germany)
Robert Hirschfeld (Germany)
Heinrich Hussmann (Germany)
Jean-Marc Jézéquel (France)
Gabor Karsai (USA)
Jana Koehler (Switzerland)
Rainer Koschke (Germany)
Thomas Kühne (New Zealand)
Vinay Kulkarni (India)
Jochen Küster (Switzerland)
Ralf Lämmel (Germany)

Michele Lanza (Switzerland)
Michael Lawley (Australia)
Timothy C. Lethbridge (Canada)
Ed Merks (Canada)
Birger Møller-Pedersen (Norway)
Ana Moreira (Portugal)
Pierre-Alain Muller (France)
Richard Paige (UK)
Alexander Pretschner (Switzerland)
Gianna Reggio (Italy)
Bernhard Rumpe (Germany)
Andy Schürr (Germany)
Bran Selic (Canada)
Perdita Stevens (UK)
Eleni Stroulia (Canada)
Gabriele Taentzer (Germany)
Juha-Pekka Tolvanen (Finland)
Laurence Tratt (UK)
Axel Uhl (Germany)
Hans Vangheluwe (Canada)
Dániel Varró (Hungary)
Eelco Visser (The Netherlands)
Markus Völter (Germany)
Andrzej Wasowski (Denmark)
Thomas Weigert (USA)
Jon Whittle (UK)

Expert Reviewer Panel

Aditya Agrawal (USA)
Jean-Michel Bruel (France)
Nancy Day (Canada)
Sebastian Fischmeister (Canada)
Sébastien Gérard (France)
Jeff Gray (USA)
Ileana Ober (France)
Kasper Østerbye (Denmark)
Awais Rashid (UK)
Andreas Rummler (Germany)
Peter Sesoft (Denmark)

Steering Committee

Thomas Baar (Switzerland)
Jean Bézivin (France)
Lionel Briand (Norway)
Steve Cook (UK)
Gregor Engels (Germany)
Andy Evans (UK)
Robert France (USA)
Geri Georg (USA)
Martin Gogolla (Germany)
Heinrich Hussmann (Germany)
Jean-Marc Jézéquel (France)
Stuart Kent (UK)
Cris Kobryn (USA)
Ana Moreira (Portugal)
Pierre-Alain Muller (France)
Oscar Nierstrasz (Switzerland)
Gianna Reggio (Italy)
David Rosenblum (UK)
Bernhard Rumpe (Germany)
Douglas C. Schmidt (USA)
Bran Selic (Canada)
Perdita Stevens (UK)
Jon Whittle (Chair; USA)

Sponsors

ACM Special Interest Group on Software Engineering
(www.sigsoft.org)
IEEE Computer Society
(www.computer.org)

Additional Reviewers

Vasco Amaral
Carsten Amelunxen
Malte Appeltauer
Egidio Astesiano
Richard Atterer
Andrej Bachmann
Omar Bahy Badreldin
Florence Balagtas-Fernandez
András Balogh
Jan-Christopher Bals
Olivier Barais
Bruno Barroca
Thiago T. Bartolomei
Dominikus Baur
Basil Becker
Nelly Bencomo
Alexej Beresnev
Rodrigo Bonifacio
Phil Brooke
Erwan Brottier
Achim D. Brucker
Fabian Buettner
Sebastian Cech
Maura Cerioli
Franck Chauvel
Dan Chiorean
Fabian Christ
Peter J. Clarke
Michelle Crane
Arnaud Cuccuru
Akshay Dabholkar
Duc-Hanh Dang
Vegard Dehlen
Romain Delamare
Deepak Dhungana
Zinovy Diskin
Nikos Drivalos
Cedric Dumoulin
Huascar Espinoza
Johan Fabry
Fernando Castor Filho
Franck Fleurey

Alexander Foerster
Frédéric Fondement
István Forgács
Andrew Forward
László Gönczy
Christian Gerth
Giacomo Ghezzi
Thomas Goldschmidt
Pieter Van Gorp
Emanuel Grant
Danny Groenewegen
Hans Groenniger
Iris Groher
Lindsay Groves
Roy Grønmo
Thomas Gschwind
Baris Güldali
Ulrich Hannemann
Michael Haupt
Zef Hemel
Christoph Herrmann
Anders Hessellund
Thomas Hettel
Mark Hibberd
Stephan Hildebrandt
Otmar Hilliges
Berthold Hoffmann
Jippe Holwerda
Ákos Horváth
Michael Jackson
Mikoláš Janota
Cedric Jeanneret
Jendrik Johannes
Frédéric Jouault
Christophe Jouvray
Stefan Jurack
Markus Kaiser
Sven Karol
Lennart C. L. Kats
Nima Kaviani
Amogh Kavimandan
Dae-Kyoo Kim

Felix Klar
Florian Klein
Renate Klempien-Hinrichs
Patrick Knab
Imre Kocsis
Dimitrios Kolovos
Máté Kovács
Holger Krahn
Ingolf Krueger
Mirco Kuhlmann
Uirá Kulesza
Sabine Kuske
Elodie Legros
László Lengyel
Tihamér Levendovszky
Hongzhi Liang
Jens Lincke
Alexander De Luca
Christoph Lueth
Mass Soldal Lund
Tiago Massoni
Matthew J. McGill
Tom Mens
Kirsten Mewes
Gergely Mezei
Milan Milanovic
Dragan Milicev
Parastoo Mohagheghi
Brice Morin
Pieter Mosterman
Jean-Marie Mottu
Patrick Mukherjee
Freddy Munoz
Gunter Mussbacher
Stefan Neumann
James Noble
Jon Oldevik
Sebastian Oster
Cesare Pautasso
Luis Pedro
Gilles Perrouin
Claas Pinkernell
Eduardo Piveta
Fiona Polack

Ernesto Posse
Christopher Power
Rick Rabiser
Ansgar Radermacher
Oliver Radfelder
Y. Raghu Reddy
Dirk Reiss
Marko Ribaric
Márcio de Medeiros Ribeiro
Filippo Ricca
Sebastian Richly
Matteo Risoldi
David Roethlisberger
Jens Rommel
Louis Rose
Nilabja Roy
Suman Roychoudhury
Ahmad Saifan
Yavuz Sancar
Tim Schattkowsky
Marvin Schulze-Quester
Andreas Seibel
Sagar Sen
Steven She
Carla Silva
Karsten Sohr
Arnor Solberg
Christian Soltenborn
Jean-Sebastien Sottet
Jim Steel
Mark Stein
Matthew Stephan
Gerson Sunye
Eugene Syriani
Dániel Tóth
Safouan Taha
Gergely Varró
Sander Vermolen
Hendrik Voigt
Steven Völkel
Arild Waaler
Ksenia Wahler
Michael Wahler
Ingo Weisemoeller

Christian Wende
Bernhard Westfechtel
Gerd Wierse
Michael Wuersch
Jochen Wuttke
Andreas Wübbeke
Vadim Zaytsev
Olaf Zimmermann
Steffen Zschaler

Table of Contents

Constraints

Model Analysis

Service-Oriented Architectures

Adaptive and Autonomic Systems

Empirical Studies

Evolution and Reverse Engineering

Modeling Language Semantics

Dependability Analysis and Testing

Aspect-Oriented Modeling

Structural Modeling

Embedded Systems

Workshops, Symposia, Tutorials, and Panels

Assessing the Influence of Stereotypes on the Comprehension of UML Sequence Diagrams: A Controlled Experiment

Marcela Genero[1], José A. Cruz-Lemus[1], Danilo Caivano[2], Silvia Abrahão[3], Emilio Insfran[3], and José A. Carsí[3]

[1] ALARCOS Research Group,
Department of Technologies and Information Systems,
University of Castilla-La Mancha
Paseo de la Universidad, 4
13071 Ciudad Real, Spain
{Marcela.Genero,JoseAntonio.Cruz}@uclm.es
[2] Department of Informatics
University of Bari
Via E. Orabona, 4
70126 Bari, Italy
caivano@di.uniba.it
[3] ISSI Research Group
Department of Information Systems and Computation
Universidad Politécnica de Valencia
Camino de Vera s/n
46022, Valencia, Spain
{sabrahao,einsfran,pcarsi}@dsic.upv.es

Abstract. The main goal of this paper is to provide empirical evidence, through a controlled experiment, of the influence of stereotypes when modelers, developers, and maintainers have to comprehend UML sequence diagrams. The comprehension of UML sequence diagrams with and without stereotypes was analyzed from three different perspectives: semantic comprehension, retention and transfer. The experiment was carried out with 77 fourth year undergraduate students of Computer Science from the University of Bari in Italy. The results obtained show a slight tendency in favor of the use of stereotypes in facilitating the comprehension of UML sequence diagrams, although it is not statistically significant. Further replications are needed to obtain more conclusive results.

1 Introduction

A wide use of models in software development has resulted in the creation of several modeling languages and notations with which to express graphical models. The most successful modeling language is the Unified Modeling Language (UML) [19], which is widely accepted in both industry and academia.

The main drawback of this language is that the set of general modeling constructs is not suitable for a specific purpose and hereafter the modeling might not be effective

K. Czarnecki et al. (Eds.): MoDELS 2008, LNCS 5301, pp. 280–294, 2008.

[27]. To address this drawback, several works propose the improvement of modeling with UML by customizing it with the extension mechanisms inherent in the language (e.g., stereotypes).

Stereotypes are used to clarify or extend the meaning of model elements and to introduce new modeling elements reusing the syntax of similar elements already available in the notation. However, does the use of stereotypes really lead to better models? In recent years, several studies through which to address this question have been conducted in the field of Software Engineering [24, 27]. These studies are focused on investigating the influence of stereotypes on the comprehension of UML class and collaboration diagrams. It is widely recognized that comprehension is one of the main factors to influence maintainability [9, 20, 23, 25], which must be assessed from the first steps of the software lifecycle, in the modeling phase (see related work in Section 2).

In our experience stereotypes are often used in industrial contexts and their use spans from use cases to class diagrams. Indeed industries use stereotypes within their development processes in order to specialize general processes to a particular technology in use, such as programming languages (C#, Java,...), application type (real time, web application, client-server, standalone, etc.), reusable components used (Microsoft Foundation Class Library, Enterprise Java Beans Library, etc.) or simply to give more detailed guidelines to the practitioners involved in the systems development processes. The use of stereotypes is also widespread in the academia in order to give students some practical instruments for guiding software design and to fill the information and representation gap of UML standard language. Well known stereotypes widely used in industry and academy are the Conallen's stereotypes [5] that are also supported by the most used UML Case Tool such as ARGO UML, Enterprise Architect and STAR UML. Today, these case tools are widely used in industrial contexts.

Nevertheless, the influence of stereotypes on the comprehension of requirements models, such as sequence diagrams has not yet been investigated. This fact has motivated us to develop the research presented in this work.

We focus on UML sequence diagrams since they are a widely used technique for reasoning about object interactions needed to realize a given scenario of functional requirement and on the comprehension of these diagrams since this is essential in the validation of requirements specifications among developers and stakeholders. A sequence diagram must be first comprehended before any desired changes to it can be identified, designed, or implemented.

In this paper, we present a controlled experiment to assess whether the use of stereotypes improves the comprehension of UML sequence diagrams. Specifically, we consider a set of stereotypes that have been proposed to enrich the semantics of interaction messages in UML sequence diagrams in the context of a Requirements Engineering approach for model-driven software development [10, 11]. The benefit of using these stereotypes is twofold. First, they can improve the comprehension of UML sequence diagrams, and second, they give specific information about how to deal with each individual source element in the transformation of UML sequence diagrams into conceptual models in the model-driven development process.

The controlled experiment was carried out by 77 fourth year of Computer Science undergraduate students at the University of Bari in Italy.

The structure of the paper is as follows. Firstly, related works on the empirical evaluation of UML diagram comprehension and specifically with regard to the use stereotypes are presented in Section 2. The definition of stereotypes for UML sequence diagrams is explained in Section 3. The description and execution of the experiment is described in Section 4. The data analysis and the interpretation of the experiment results are presented in Section 5. Finally, the conclusions and a discussion on future research work are presented in Section 6.

2 Related Work

The comprehension of UML diagrams has frequently been investigated (this list does not intend to be exhaustive, but simply illustrative):

- Purchase et al. [21, 22] report two experiments which take the perspective of human comprehension of UML class and collaboration diagrams considering five different notations in each study. In the case of class diagrams, they obtained that the best performing notation depends on the task for which the diagram was used, whilst for the collaboration diagrams, they concluded that a specification matches a diagram, but not when errors in a diagram can be identified.
- Cruz-Lemus *et al.* [6] present a family of empirical studies to investigate whether the use of composite states improves the understandability of UML statechart diagrams. The results, although not conclusive, show that the use of composite states does not improve the understandability of simple UML statechart diagrams.
- Genero *et al.* [9] present a family of experiments carried out to evaluate the relation between a set of measures for the structural properties of UML class diagrams and their understandability and modifiability. They also provide prediction models for such quality characteristics. They found that the measures related to associations and generalization seems to have more influence on the understanding and modification of UML class diagrams.
- Lange and Chaudron [15] use an experiment to validate whether there is a difference between a set of views which they propose to increase model understanding (MetaView, ContextView, MetricView, and UML-City-View), and the existing views with respect to comprehension correctness and effort. They conclude that correctness is improved and time needed is reduced when the proposed views are used.
- Xie *et al.* [30] present a study in which their proposed synchronization adorned UML sequence diagrams notation is used to verify whether it helps students towards a better understanding of concurrent executions and concurrency concepts. They finally find a statistically significant benefit.

To our knowledge, there are two recent works which are aimed at investigating the influence of stereotypes on the comprehension of UML diagrams:

- Staron *et al.* [27] have assessed the influence of stereotypes on the comprehension of models (UML class and collaboration diagrams) through a set of experiments. They found that stereotypes significantly helped both students and industrial programmers.

- Ricca *et al.* [24] describe a family of three experiments which were performed to assess the effectiveness of UML stereotypes in supporting comprehension tasks in class diagrams for Web design. They conclude that stereotypes significantly improve the performance of low ability/experience subjects.

The literature review described here reveals that comprehension is a principal concern in the context of UML modeling, but that the contribution of stereotypes with which to improve the comprehension of UML sequence diagrams has not yet been investigated.

3 Stereotypes for UML Sequence Diagrams

UML indicates that sequence diagrams are a means to model an aspect of the dynamic behavior of a system [12]. They can be used in the context of the whole system, of a subsystem, or they can be attached to a Use Case. Some authors indicate that at least one sequence diagram should be drawn per Use Case [7, 12]. When a sequence diagram is developed for a Use Case, the Use Case description can be used to develop at least the initial draft of the sequence diagram. Throughout the design process the Use Case diagram can be revised based on the results of the sequence diagram, and vice versa, until both models are appropriately tuned [16].

When using a sequence diagram at the type level, it represents a pattern interaction, which is a set of messages among object types with which to carry out a behavior. It has two dimensions: the vertical dimension represents time, and the horizontal dimension represents the different object types (classes). Time proceeds down the page and there is no significance to the horizontal ordering of the object types.

The emphasis in this type of diagrams is that of graphically representing the pattern interaction between object types by sending and receiving messages as time advances. In addition, when using sequence diagrams to realize a Use Case, the focus should not be on specifying detailed behavior with complex iterations and conditional messages (the *how*) but the interactions that are needed to perform the Use Case purpose (the *what*).

There is at least one sequence diagram per Use Case, one for the basic course of action, and another for each alternative course of action (if any). To build the sequence diagram for a Use Case, we analyze each Use Case at two levels:

- **Use Case diagram level:** actors that communicate with the Use Case;
- **Use Case specification level:** the set of steps or responsibilities to be performed in order to accomplish the Use Case.

At the Use Case diagram level, an actor can send and/or receive information to/from the Use Case. Every Use Case has at least one actor, which is the Use Case initiator and other collaborating actors. All the actors are potentially object types to be taken into account for the sequence diagram.

At the Use Case specification level, the main task is to obtain the set of responsibilities implied by the corresponding Use Case steps. Essentially, when the Use Case receives a stimulus (an external interaction), the system produces a set of interactions between its internal components (objects) as a response. These interactions are represented as messages in the sequence diagram. In [10, 11] they are classified according to their nature by using the following UML stereotypes:

- **«signal»** this is a message between an actor and the system.
- **«service»** this is a message that updates the state of an object. The properties new/destroy/update can be used if the object is to be created, destroyed or modified.
- **«query»** this is a message to query the state of a specific object or set of objects.
- **«connect»** this is a message which is used to indicate that an object from the sender object type needs to be connected (or related) to an object from the receiver object type.

Although the UML 2.1 standard [19] defines a set of general predefined stereotypes that can be used in UML depending on the L2 or L3 level of compliance, there are no specific stereotypes for sequence diagrams. For this reason, specific methods or tools should extend the notation and meaning provided by the standard UML features if they are to have a semantically richer modeling notation. In Figure 1, we present an example which shows the stereotyped interactions (messages) that are needed to realize a Use Case for items-sale in a conventional store. The explanation of the main interaction/messages is also given below.

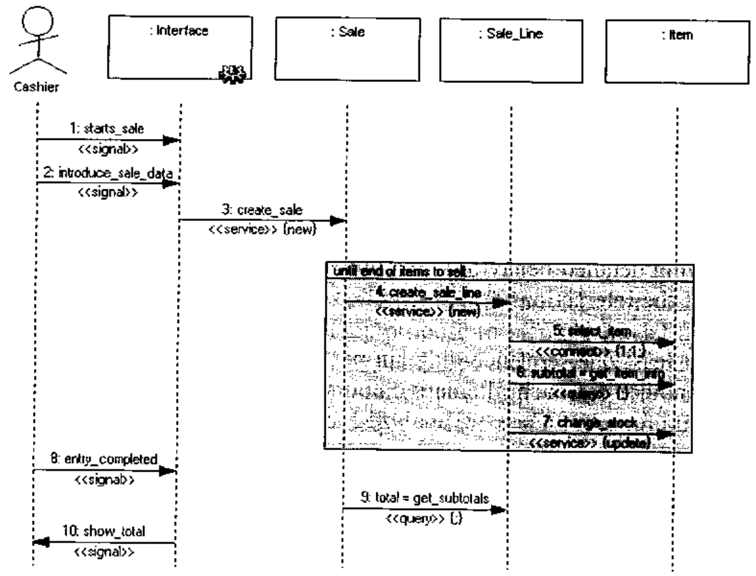

Fig. 1. An example of a sequence diagram with stereotyped messages

- **Message 1 and 2.** The cashier starts a sale. This message represents the commencement of communication between the Cashier and the Interface of the software system. The following message *introduce_sale_data* provides the data which is necessary to perform the operation.

- **Message 3.** The cashier creates a new sale. This message implies the responsibility of creating a new sale and registering its information. We have named this responsibility *create_sale* and allocated it to a new class called Sale.
- **Message 4.** The cashier records the identifier of each item to be sold. This action implies the responsibility of registering information about the items to be sold. We have named this responsibility *create_sale_item* and allocated it to a new class component called Sale-line.
- **Message 5.** The message *select_one_item* (with the stereotype **«connect»**) is identified because the analyst determines that a complete Sale-line must have one and only one Item (product) assigned to it.
- **Message 6.** The system determines the item price and adds this information to the current *sale_line*. This action implies the responsibility of querying an object that stores the item price. We have named this responsibility *get_item_info* with the stereotype **«query»**.
- **Message 7.** Another responsibility of the Use Case is that of updating the item stock when the item is sold. We have named this responsibility *change_stock* and allocated it to the previously detected class component Item.
- **Message 9.** The system calculates the total amount of the sale from the corresponding set of *sale_lines* and the system then shows the information concerning the complete sale to the Cashier (Message 10).

Note that at this level of specification, the details of the iteration or the control flow in the sequence diagram are not the main focus since we are mainly identifying and specifying *what* interactions are needed to realize the Use Case. This activity leads us to discovering classes and services and not to describing the complete behavior of the involved classes.

The process of specifying a Use Case as an interaction of collaborative objects continues thus until all the steps of the Use Case are defined in their corresponding sequence diagram. Developers should verify this specification looking for the satisfaction of the Use Case purpose and also validate it together with users during and after its construction. Comprehension is, therefore, a critical factor in the building of this type of diagrams.

4 Experiment Description

The main goal of this experiment is to assess whether the use of stereotypes improves the understanding of UML sequence diagrams. Therefore, by using the GQM template for goal definition [1, 3], the goal of our experiment is defined as follows:

"Analyze the use of stereotypes for the purpose of evaluating it with respect to the comprehension of UML sequence diagrams from the point of view of the researcher, in the context of fourth year undergraduate students in Computer Science from the University of Bari".

In order to run and report this experiment we followed the recommendations provided in several works [13, 29]. The design of the experiment presented in this paper is similar to those presented in [27].

We selected a balanced factorial design with group-interaction confounding [14]. The objects were UML sequence diagrams, with two possible values – stereotyped diagram and non-stereotyped diagram.

All the subjects were randomly assigned to 4 groups (1, 2, 3 and 4). The experiment consisted of two rounds. Two different diagrams were presented to every subject in each group. To avoid a possible learning effect, the diagrams came from different application domains (A and B). All diagrams and domains were similar in complexity and similarly common to all the subjects. More details about the experimental design are provided in the remainder of this section.

4.1 Planning

Various issues related to the planning of the experiment are introduced in this section.

- **Subjects.** The experiment was carried out by 77 fourth year Computer Science students who were taking a course in Empirical Software Engineering. The subjects had participated in two Software Engineering courses in which they had acquired training in UML diagrams. Their knowledge was sufficient for them to understand the given non-stereotyped diagrams, and they had roughly the same background. They had knowledge about the use of stereotypes in general, but they were taught about the stereotypes we proposed for UML sequence diagrams in a training session organized to take place the day before the experiment was carried out.

To avoid social threats due to evaluation apprehension, the students were not graded on their performance. On the contrary, the participants were granted 2 extra points in their final valuation at the end of the course

- **Experiment objects.** The experimental objects consisted of four diagrams, summarized below (these are presented fully at alarcos.esi.uclm.es/ExpStereotypes):

- DA-S: stereotyped diagram A and a general description of each type of stereotype.
- DA-N: non-stereotyped diagram A.
- DB-S: stereotyped diagram B and a general description of each type of stereotype.
- DB-N: non-stereotyped diagram B.

UML sequence diagram A-x describes a car rental domain. It describes an "Extras Rental" use case which may occur in addition to the car rental. Diagram B-x describes a hotel domain. It represents the sequence of interactions of the "Book room" use case. These experimental objects were presented in Italian.

- **Independent and dependent variables.** There are two independent variables in the experiment, the diagram type, with values: S (stereotyped) and N (non-stereotyped), and the diagram domain (A and B). By combining each level of the independent variables we obtain four treatments, as reflected by the four diagrams which are the objects of the experiment.

In accordance with certain suggestions concerning the measurement of comprehension [4, 6, 8], we have used the Cognitive Theory of Multimedia Learning (CTML) [18]. This choice was made on the basis of several reasons. First, it focuses on words and graphics, which are the elements in the UML sequence diagrams grammar. Second, it provides principles for the design of effective multimedia presentations which

can be tested empirically. Third, it has evolved through years of work and development of experimental instruments and methods related to model comprehension [17, 18].

By following CTML, the comprehension of the UML sequence diagrams has been defined through three variables:

- **Semantic comprehension:** the ability to comprehend the semantics of the models.
- **Retention:** the comprehension of material being presented, and the ability to retain knowledge from it.
- **Transfer:** the ability to use the knowledge gained from the material to solve related problems which are not directly answerable from it.

It is expected that due to the use of stereotypes that enrich the meaning of the messages in the sequence diagram specification, the reader of a stereotyped sequence diagram will have additional key knowledge than a reader of a non-stereotyped one for comprehension, retention and transfer tasks.

For clearness, the quality model used (defined according to the GQM approach [1, 3]) is presented in Figure 2, which also points out the experimental hypothesis defined for each measure used.

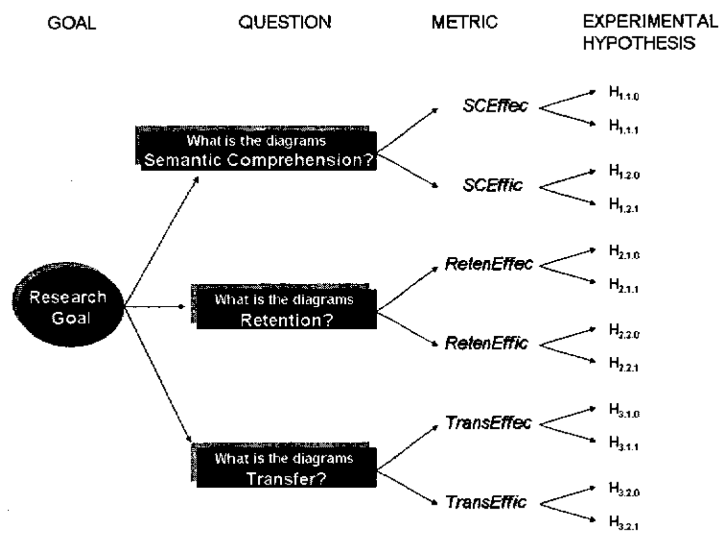

Fig. 2. Quality Model used and Experimental Hypotheses

In order to measure these variables, we have used three separate tests based on questionnaires. Each comprehension measure is computed as:

- Effectiveness: the proportion of correct answers provided in each test (number of correct answers / number of questions). This measure reflects the ability to understand the presented material correctly.
- Efficiency: the proportion of correct answers divided by time (Effectiveness / Time).

In this work, we call these values SCEffec/SCEffic (Semantic Comprehension Effectiveness and Efficiency), TransEffec/TransEffic (Transfer Effectiveness and Efficiency), and finally, RetenEffec/RetenEffic (Retention Effectiveness and Efficiency).

- **Hypotheses.** For Semantic Comprehension we wished to test the following hypothesis:
- Null hypotheses
  $H_{1.1,0}$: stereotypes do not improve the subjects' SCEffec when attempting to comprehend a UML sequence diagram.
  $H_{1.2,0}$: stereotypes do not improve the subjects' SCEffic when attempting to comprehend an UML sequence diagram
- Alternative hypotheses:
  $H_{1.1,1} = \neg H_{1.1,0}$
  $H_{1.2,1} = \neg H_{1.2,0}$

We analogously formulated a set of hypothesis H_2 for Retention measures (RetenEffec and RetenEffic), and another set H_3 related to the Transfer measures (TransfEffec and TransEffic) (see Figure 2).

We stated one-tailed hypotheses as we wanted to check if the use of stereotypes improved the way in which subjects comprehended UML sequence diagrams.

- **Instrumentation.** The instruments used in the experiment were three tests attached to each of the four treatments. In each test the subjects were asked to write down the time before starting to solve the tasks required in each test.
  The tests are described as follows:
- Test 1: contained 10 Yes/No questions concerning the semantic of the diagrams. This task was used to obtain SCEffec and SCEffic.
- Test 2: consisted of a 'fill-in-the-blanks' task in which the subjects had to complete a test describing the functionality of the diagrams. This task was used to calculate RetenEffec and RetenEffic.
- Test 3: the subjects had to name a set of new messages that had been attached to the original version of the diagrams, but were labeled only with the parameters. This task was used to calculate TransEffec and TransEffic.

4.2 Experiment Operation

The experiment consisted of two rounds. In each round, each of the groups was given a different treatment. In order to alleviate learning effect, as is previously mentioned, we divided the experimental subjects into 4 groups. We randomly assigned the corresponding diagrams to each group, but in different order. Table 1 presents the outline of the experimental operation. The description of each type of diagram has already been presented in section 4.1. This assignment of diagrams corresponds with the selected balanced factorial design with group-interaction confounding [14], which permits the alleviation of learning and fatigue effects.

Table 1. Experiment rounds

Round 1		Diagram Type S	Diagram Type N
Domain	A	Group 1	Group 2
	B	Group 4	Group 3

Round 2		Diagram Type S	Diagram Type N
Domain	A	Group 3	Group 4
	B	Group 2	Group 1

The design of the experiment includes a previous three hour lecture which took place the day before the experiment was carried out. The lecture included various tasks:

- To introduce the students the stereotypes defined for UML sequence diagrams.
- To show the students an example, similar to the material used in the experiment. We solved the tasks in the example with the students.
- To perform a pilot study in order to assure that the students had understood the experimental material and tasks.

After this previous session, we checked the responses of the students, and we discovered that the students performed badly when solving the second task. Therefore, before the experiment was carried out, we carefully re-explained the tasks involved in Test 2.

4.3 Conducting the Experiment

The day after the training session, the experiment took place in a one and a half hour session. In the 30 first minutes we explained how to perform the experiment and randomly assigned the subjects to two balanced groups.

The experiment was conducted in a classroom, where the students were supervised and no communication among them was allowed. Both groups were located in the same room.

Each round (see Table 1) was performed in the following way:

1. The subjects received the material for Test 1 which included a UML sequence diagram and a questionnaire. After finishing this task, Test 1 was handed back to the supervisor.
2. The students received Test 2, which had to be solved without the UML sequence diagram. After completing the tasks, Test 2 was handed back to the supervisor.
3. The students received Test 3 and after solving it they returned it to the supervisor.

5 Data Analysis and Interpretation

All the statistical analyses presented in this section were performed using SPSS [26].

In this section, we present, for each variable, the descriptive statistics and the results of the ANOVA tests carried to test the formulated hypotheses. ANOVA is the most appropriate test with which to explore the results of a factorial design with interaction confounded [14, 28], i.e., the design used in our experiment.

All the tests in our study were carried out by using a statistical significance threshold of $\alpha = 0.05$. The null hypotheses were thus rejected if the statistical tests we used provided a statistical significance of the results that was not higher than 0.05.

Tables 2, 3, and 4 presents the descriptive statistics for the all the variables studied: Semantic comprehension, Retention and Transfer. Grey cells indicate the best subjects' performance.

Table 2. Descriptive statistics for the Semantic Comprehension

Diagram Type	Domain	N	SCEffec Mean	SCEffec Std. Dev	SCEffic Mean	SCEffic Std. Dev
S	A	38	0.756	0.1410	0.0020	0.00069
	B	39	0.782	0.1312	0.0027	0.00133
	Total	77	0.769	0.1360	0.0023	0.00112
N	A	39	0.761	0.1079	0.0022	0.00082
	B	38	0.741	0.1390	0.0024	0.00096
	Total	77	0.751	0.1242	0.0023	0.00089

For both the variables (SCEffec and SCEffic), the subjects obtained better global results in the stereotyped diagrams. Obviously this means that the stereotyped diagrams were more comprehensible, less error prone and required less time to understand than non- stereotyped diagrams.

Table 3. Descriptive statistics for the Retention

Diagram Type	Domain	N	RetenEffec Mean	RetenEffec Std. Dev	RetenEffic Mean	RetenEffic Std. Dev
S	A	38	0.636	0.1579	0.0029	0.0036
	B	39	0.784	0.1830	0.0035	0.0019
	Total	77	0.709	0.1854	0.0032	0.0029
N	A	39	0.637	0.1814	0.0024	0.0011
	B	38	0.807	0.1408	0.0036	0.0044
	Total	77	0.723	0.1823	0.0030	0.0032

Table 4. Descriptive statistics for the Transfer

Diagram Type	Domain	N	TransEffec Mean	TransEffec Std. Dev	TransEffic Mean	TransEffic Std. Dev
S	A	38	0.682	0.2501	0.0023	0.0011
	B	39	0.829	0.2404	0.0090	0.0058
	Total	77	0.755	0.2547	0.0056	0.0053
N	A	39	0.616	0.2248	0.0022	0.0013
	B	38	0.821	0.3139	0.0097	0.0064
	Total	77	0.719	0.2907	0.0060	0.0059

Table 3 shows that, related to Retention, the subjects obtained better results in effectiveness when they did not use stereotypes, while the scores were higher for efficiency when they worked with stereotyped diagrams. In addition, RetenEffec and RetenEffic values are significantly better for the diagrams of domain B. Therefore, the subjects who used non-stereotyped diagrams were able to recall a greater number of

correct functionalities of the models, but the subjects with non-stereotyped diagrams performed better if we relate the number of correct answers with the time spent answering the test. This is probably due to the fact that stereotyped diagrams contain more information than those which are non- stereotyped and thus prove to be more difficult to remember (and not to understand!). It is interesting to note that effectiveness is calculated as the number of correct answers divided by the total number of expected answers. In our analysis, the unanswered questions are, at present, considered to be wrong answers. However, it is probable that if we consider effectiveness as the number of correct answers divided by the total number of given answers, the effectiveness would increase. The students were told not to fill in the blank if they did not remember the answer. The higher efficiency for the stereotyped diagrams seems to confirm this interpretation: the students answered what they remembered in a short time and did not consider the rest of answers.

The results for Transfer (Table 4) are the opposite of those for Retention. Better results were obtained for effectiveness when the subjects used the stereotyped diagrams, but they were worse for efficiency. Moreover, these results confirm that the stereotyped diagrams are more comprehensible and less error prone but that the quantity of information they include implies that more time is needed for diagram processing: in this case, and unlike the Semantic Comprehension, the students have to reconstruct the message flow in the sequence diagram and thus have to spend more time in understanding the semantic first and the syntactic of the diagrams later. This implies more time spent, which thus leads to lower efficiency.

Table 5 presents the results of the ANOVAs performed to check the hypothesis presented in Section 4.1. From this table, we can observe that in most cases, the domain significantly affects the comprehension of the diagrams (see bold cells). Table 2 evidences that the subjects performed worse with the diagrams of domain A (car rental). This could imply that the subjects found the car rental domain more difficult.

With respect to the use of stereotypes, there are no significant values in the table. Moreover, the powers of the tests, i.e. the ability to detect an effect if there is one, are low. This implies that we cannot reject the null hypotheses investigated in this work and accepting the alternative hypotheses could be risky and misleading. Furthermore we have results that might be a consequence of the treatment or be determined by an interaction effect between treatment and domain.

A similar data analysis which considered the data of each round separately was also carried out. The findings were worse. The order effect was also studied statistically, despite its being alleviated with the selected experimental design. As it was expected, no effect was found.

Table 5. ANOVAs results

Effectiveness		SCEffec			RetenEffec			TransEffec		
Source	Df	F	sig.	Power	F	sig.	Power	F	sig.	Power
Stereotypes	1	0.749	0.388	0.138	0.206	0.650	0.074	0.798	0.373	0.144
Domain	1	0.018	0.893	0.052	35.369	**0.000**	1.000	17.680	**0.000**	0.987
Efficiency		**SCEffic**			**RetenEffic**			**TransEffic**		
Source	Df	F	sig.	Power	F	sig.	Power	F	sig.	Power
Stereotypes	1	0.311	0.578	0.086	0.126	0.723	0.064	0.157	0.693	0.068
Domain	1	9.422	**0.003**	0.862	3.386	0.068	0.448	98.538	**0.000**	1.000

5.1 Threats to Validity

We must consider certain issues which may have threatened the validity of the experiment:

- Conclusion validity: The random heterogeneity of subjects is always present when experimenting with students and we are also conscious that they had no previous knowledge of the stereotypes used in the UML sequence diagram included in the experimental materials. Furthermore, if the knowledge of the students involved in the experiment could be assumed to be comparable to that of junior industry professionals, the working pressure and the overall environment in industry is different. We assumed a subjects' homogenous background, but this was not evaluated before the experiment was carried out.
- Construct validity: The hypothesis of equivalence between the complexities of the domains was not confirmed by data analysis and this has been one of the pitfalls of this work. In fact, there were relevant differences in the subject performances which can be attributed to the differences in the diagram domains: the car rental diagram was found to be much more difficult than that of the hotel.

6 Conclusions and Future Work

The main concern of this paper is to investigate the use of stereotypes in the context of UML sequence diagram from three different perspectives of the comprehension of UML sequence diagrams: Semantic Comprehension, Retention, and Transfer. This took place through a controlled experiment and, unfortunately the obtained results cannot be assumed to be final.

The p-values observed in the hypotheses test are non-significant and thus the null-hypotheses cannot be rejected. This implies that the stereotypes do not significantly improve comprehension in a statistical sense. Nevertheless, in 4 out of 6 cases the scores obtained by the subjects were higher when using stereotypes, which could imply that the use of stereotypes does, to some extent, help in the diagram comprehension.

Obviously these conflicting elements stimulate further investigation which we wish to carry out in the near future. We shall attempt to control the threats in a more effective manner and improve the experimental schema, addressing the following issues:

- The Domain effect must be eliminated before conclusions with regard to domains with a similar level of difficulty are drawn, or perhaps, a single domain should be used.
- We must confront the experience factor by attempting to point out the effect that the experience of using our stereotypes had on the subjects' performance. One possible way in which to do this might be to evaluate experience before the experiment and to consequently form the groups on the basis of blocking by experience;
- Data analysis must be improved by clarifying the difference between the non unanswered questions and the wrong answers and their impact on Effectiveness and Efficiency. In this work they are dealt with in the same way.

- According to Rica et al. [24], it may be advisable to investigate the possible existence of other relevant factors. This will be done by extending the definition of the actual quality model in use.

The present work is the first experiment within a planned family of experiments, and it considers not only students but also practitioners. In fact, as Basili et al. [2] note, families of studies are needed for knowledge building. We are already planning to replicate this experiment with fifth year students in Computer Science at the University of Castilla-La Mancha. The experience collected in this first study and in that of Castilla-La Mancha will allow us to refine the material and tasks with the objective of choosing the optimal configuration of the experiment and will make it possible to perform a case study in an industrial setting.

Acknowledgements

This research is part of the META project (TIN2006-15175-C05-05) financed by the "Ministerio de Educación y Ciencia (Spain)", the IDONEO project (PAC08-0160-6141) financed by "Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha", and the Quality-driven Model Transformations project financed by the Universidad Politécnica de Valencia.

The authors would like to thank professors Giuseppe Visaggio and Maria Teresa Baldassarre (University of Bari) and their students for having cooperated in the performance of the experiment.

References

[1] Basili, V., Rombach, H.: The TAME Project: Towards Improvement-oriented Software Environments. IEEE Transactions on Software Engineering 14(6), 758–773 (1988)
[2] Basili, V., Shull, F., Lanubile, F.: Building Knowledge through Families of Experiments. IEEE Transactions on Software Engineering 25, 456–473 (1999)
[3] Basili, V., Weiss, D.: A Methodology for Collecting Valid Software Engineering Data. IEEE Transactions on Software Engineering 10(6), 728–738 (1984)
[4] Bodart, F., Patel, A., Sim, M., Weber, R.: Should Optimal Properties Be Used in Conceptual Modelling? Theory and Three Empirical Tests. Information Systems Research 12(4), 384–405 (2001)
[5] Conallen, J.: Building Web Applications with UML. Addison-Wesley Publishing Company, Reading (2000)
[6] Cruz-Lemus, J.A., Genero, M., Morasca, S., Piattini, M.: Using Practitioners for Assessing the Understandability of UML Statechart Diagrams with Composite States. In: Hainaut, J.-L., Rundensteiner, E.A., Kirchberg, M., Bertolotto, M., Brochhausen, M., Chen, Y.-P.P., Cherfi, S.S.-S., Doerr, M., Han, H., Hartmann, S., Parsons, J., Poels, G., Rolland, C., Trujillo, J., Yu, E., Zimányie, E. (eds.) ER Workshops 2007. LNCS, vol. 4802, pp. 213–222. Springer, Heidelberg (2007)
[7] Eriksson, H.E., Penker, M.: UML Toolkit. John Wiley and Sons, Chichester (1998)
[8] Gemino, A., Wand, Y.: Complexity and Clarity in Conceptual Modeling: Comparison of Mandatory and Optional Properties. Data and Knowledge Engineering 55, 301–326 (2005)

[9] Genero, M., Manso, M.E., Visaggio, A., Canfora, G., Piattini, M.: Building Measure-Based Prediction Models for UML Class Diagram Maintainability. Empirical Software Engineering 12, 517–549 (2007)
[10] Insfran, E.: A Requirements Engineering Approach for Object-Oriented Conceptual Modeling. In: DSIC 2003, University of Technology of Valencia
[11] Insfran, E., Pastor, P., Wieringa, R.: Requirements Engineering-based Conceptual Modeling. Journal of Requirements Engineering 7(2), 61–72 (2002)
[12] Jacobson, I., Christerson, M., Jonsson, P., Overgaard, G.: Object-Oriented Software Engineering, a Use Case-driven Approach. Addison-Wesley, Reading (1992)
[13] Juristo, N., Moreno, A.: Basics of Software Engineering Experimentation. Kluwer Academic Publishers, Dordrecht (2001)
[14] Kirk, R.E.: Experimental Design: Procedures for the Behavioral Sciences, 3rd edn. Brooks/Cole Publishing Company (1995)
[15] Lange, C.F.J., Chaudron, M.R.V.: Interactive Views to Improve the Comprehension of UML Models - An Experimental Validation. In: Proceedings of 15th IEEE International Conference on Program Comprehension (ICPC 2007), Banff, Canada, pp. 221–230 (2007)
[16] Larman, C.: Applying UML and Patterns. Prentice-Hall, Englewood Cliffs (1998)
[17] Mayer, R.E.: Models for Understanding. Review of Educational Research 59(1), 43–64 (1989)
[18] Mayer, R.E.: Multimedia Learning. Cambridge University Press, Cambridge (2001)
[19] OMG, UML 2.1 Unified Modeling Language (TM) (2006)
[20] Otero, M.C., Dolado, J.J.: Evaluation of the Comprehension of the Dynamic Modeling in UML. Information and Software Technology 46(1), 35–53 (2004)
[21] Purchase, H.C., Colpoys, L., McGill, M., Carrington, D.: UML Collaboration Diagram Syntax: an Empirical Study of Comprehension. In: Proceedings of 1st International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT 2002), Paris, France, pp. 13–22 (2002)
[22] Purchase, H.C., Colpoys, L., McGill, M., Carrington, D., Britton, C.: UML Class Diagram Syntax: an Empirical Study of Comprehension. In: Proceedings of Australian Symposium on Information Visualisation, Sydney, Australia, pp. 113–120 (2001)
[23] Reinhartz-Berger, H., Dori, D.: OPM vs. UML - Experimenting with Comprehension and Construction of Web Application Models. Empirical Software Engineering 10, 57–79 (2005)
[24] Ricca, F., Di Penta, M., Torchiano, M., Tonella, P., Ceccato, M.: The Role of Experience and Ability in Comprehension Tasks supported by UML Stereotypes. In: Proceedings of 29th International Conference on Software Engineering (ICSE 2007), Minneapolis, USA, pp. 375–384 (2007)
[25] Selic, B.: The Pragmatics of Model-Driven Development. IEEE Software 20(5), 19–25 (2003)
[26] SPSS, SPSS 12.0, Syntax Reference Guide, Chicago, USA. SPSS Inc. (2003)
[27] Staron, M., Kuzniarz, L., Wohlin, C.: Empirical Assessment of Using Stereotypes to Improve Comprehension of UML Models: a Set of Experiments. The Journal of Systems and Software 79, 727–742 (2006)
[28] Winer, B.J., Brown, D.R., Michels, K.M.: Statistical Principles in Experimental Design. McGraw-Hill, New York (1991)
[29] Wohlin, C., Runeson, P., Hast, M., Ohlsson, M.C., Regnell, B., Wesslen, A.: Experimentation in Software Engineering: an Introduction. Kluwer Academic Publisher, Dordrecht (2000)
[30] Xie, S., Kraemer, E., Stirewalt, R.E.K.: Empirical Evaluation of a UML Sequence Diagram with Adornments to Support Understanding of Thread Interactions. In: Proceedings of 15th IEEE International Conference on Program Comprehension (ICPC 2007), Banff, Canada, pp. 123–134 (2007)