

Frank Buschmann
Alejandro P. Buchmann
Mariano A. Cilia (Eds.)

LNCS 3013

Object-Oriented Technology

ECOOP 2003 Workshop Reader

ECOOP 2003 Workshops
Darmstadt, Germany, July 2003
Final Reports

ECOOP
DARMSTADT GERMANY

 Springer

Frank Buschmann Alejandro P. Buchmann
Mariano A. Cilia (Eds.)

Object-Oriented Technology

ECOOP 2003 Workshop Reader

ECOOP 2003 Workshops
Darmstadt, Germany, July 21-25, 2003
Final Reports

 Springer

Volume Editors

Frank Buschmann
Siemens Corporate Technology, Germany
Otto-Hahn-Ring 6, 81739 Munich, Germany
E-mail: Frank.Buschmann@siemens.com

Alejandro P. Buchmann
Mariano A. Cilia
Darmstadt University of Technology, Germany
FB Informatik, Databases and Distributed Systems
Hochschulstr. 10, 64289 Darmstadt, Germany
E-mail: buchmann@informatik.tu-darmstadt.de
cilia@dvs1.informatik.tu-darmstadt.de

Preface

This volume represents the seventh edition of the ECOOP Workshop Reader, a compendium of workshop reports from the 17th European Conference on Object-Oriented Programming (ECOOP 2003), held in Darmstadt, Germany, during July 21–25, 2003.

The workshops were held during the first two days of the conference. They cover a wide range of interesting and innovative topics in object-oriented technology and offered the participants an opportunity for interaction and lively discussion. Twenty-one workshops were selected from a total of 24 submissions based on their scientific merit, the actuality of the topic, and their potential for a lively interaction. Unfortunately, one workshop had to be cancelled.

Special thanks are due to the workshop organizers who recorded and summarized the discussions. We would also like to thank all the participants for their presentations and lively contributions to the discussion: they made this volume possible. Last, but not least, we wish to express our appreciation to the members of the organizing committee who put in countless hours setting up and coordinating the workshops.

We hope that this snapshot of current object-oriented technology will prove stimulating to you.

October 2003

Frank Buschmann
Alejandro Buchmann
Mariano Cilia

Library of Congress Control Number: 2004108314

CR Subject Classification (1998): D.1-3, H.2, F.3, C.2, K.4, J.1

ISSN 0302-9743
ISBN 3-540-22405-X Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable to prosecution under the German Copyright Law.

Springer-Verlag is a part of Springer Science+Business Media
springeronline.com

© Springer-Verlag Berlin Heidelberg 2004
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 11299011 06/3142 543210

Table of Contents

Exception Handling in Object Oriented Systems: Towards Emerging Application Areas and New Programming Paradigms	1
<i>Alexander Romanovsky, Christophe Dony, Anand Tripathi, Jørgen Lindskov Knudsen</i>	
Parallel Object-Oriented Scientific Computing Today	11
<i>Kei Davis, Jörg Striegnitz</i>	
Communication Abstractions for Distributed Systems	17
<i>Antoine Beugnard, Ludger Fiege, Robert Filman, Eric Jul, Salah Sadou</i>	
.NET: The Programmer's Perspective	30
<i>Hans-Jürgen Hoffmann</i>	
Component-Oriented Programming	34
<i>Jan Bosch, Clemens Szyperski, Wolfgang Weck</i>	
Workshop for PhD Students in Object Oriented Programming	50
<i>Pedro J. Clemente, Miguel A. Pérez, Sergio Lujan, Hans Reiser</i>	
Formal Techniques for Java-Like Programs	62
<i>Susan Eisenbach, Gary T. Leavens, Peter Müller, Arnd Poetzsch-Heffter, Erik Poll</i>	
Object-Oriented Reengineering	72
<i>Serge Demeyer, Stéphane Ducasse, Kim Mens, Adrian Trifu, Rajesh Vasa, Filip Van Rysselberghe</i>	
Mobile Object Systems: Resource-Aware Computation	86
<i>Ciarán Bryce, Crzegorz Czajkowski</i>	
Quantitative Approaches in Object-Oriented Software Engineering	92
<i>Fernando Brito e Abreu, Mario Piattini, Geert Poels, Houari A. Sahraoui</i>	
Composition Languages	107
<i>Markus Lumpe, Jean-Guy Schneider, Bastiaan Schönhage, Markus Bauer, Thomas Gensler</i>	
Tools and Environments for Learning Object-Oriented Concepts	119
<i>Isabel Michiels, Jürgen Börstler, Kim B. Bruce, Alejandro Fernández</i>	

Patterns in Teaching Software Development	130
<i>Erzsébet Angster, Joseph Bergin, Marianna Sipos</i>	
Object-Oriented Language Engineering for the Post-Java Era	143
<i>Wolfgang De Meuter, Stéphane Ducasse, Theo D'Hondt, Ole-Lehrman Madsen</i>	
Analysis of Aspect-Oriented Software	154
<i>Jan Hannemann, Ruzanna Chitchyan, Awais Rashid</i>	
Modeling Variability for Object-Oriented Product Lines	165
<i>Matthias Riebisch, Detlef Streitferdt, Ilan Pashov</i>	
Object Orientation and Web Services	179
<i>Anthony Finkelstein, Winfried Lamerdorf, Frank Leyman, Giacomo Piccinelli, Sanjiva Weerawarana</i>	
Advancing the State of the Art in Run-Time Inspection	190
<i>Robert Filman, Katharina Mehner, Michael Haupt</i>	
Aliasing, Confinement, and Ownership in Object-Oriented Programming	197
<i>Dave Clarke, Sophia Drossopoulou, James Noble</i>	
Author Index	209

Quantitative Approaches in Object-Oriented Software Engineering

Fernando Brito e Abreu¹, Mario Piattini², Geert Poels³, and Houari A. Sahraoui⁴

¹QUASAR Research Group, Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Monte da Caparica, Portugal

²ALARCOS Research Group, Departamento de Informática, Universidad de Castilla-La Mancha, Ciudad Real, Spain

³Department of Management Information, Operations Management and Technology Policy, Faculty of Economics and Business Administration, Ghent University, Ghent, Belgium and Centre for Industrial Management, Katholieke Universiteit Leuven, Leuven, Belgium

⁴Département d'Informatique et Recherche Opérationnelle, Université de Montréal, Montréal, Quebec, Canada

Abstract. The QAOOSE'2003 workshop brought together, for a full day, researchers and practitioners working on several aspects related to quantitative evaluation of software artifacts developed with the object-oriented paradigm. Ideas and experiences were shared and discussed. This report includes a summary of the technical presentations and subsequent discussions raised by them. Eleven out of twelve submitted position papers were presented, covering different aspects such as metrics formalization, new metrics (for coupling, cohesion, constraints or dynamic behavior) or metrics validation, to name a few. In the closing session the participants were able to discuss open issues and challenges arising from researching in this area, as well as they tried to forecast which will be the hot topics for research in the short to medium term.

1 Historical Background and Motivation

Throughout several editions¹, the QAOOSE series of workshops has attracted participants, mostly from academia and research centers somehow related to industry, which were involved or interested in the application of quantitative methods in object-oriented software engineering research and practice. Quantitative approaches in the realm of the object-oriented paradigm is a broad but active research area that aims at the development and evaluation of numerical methods, techniques, tools and practical guidelines to assess, forecast and improve the quality of software products and the efficiency and effectiveness of software processes.

2 Workshop Overview

Since this a quantitative workshop, let us start with some numbers. Seventeen (17) people, out of thirty-three (33) overall participants (if we include the co-authors that were not present) attended the workshop. They were representing eighteen (18)

¹ Information on those previous editions can be found in the end of this report.

different organizations from ten (10) different countries. Among the attendants, five (5) people were not authors, as it is normally the case in these kind of closed² workshops. They have asked the organizers to attend the workshop, which is an additional evidence of the interest raised by this area, as well as an additional measure of success of this year's edition.

This workshop encompassed four 90-minute sessions, each chaired by one of the four organizers, with the exception of Mario Piattini that asked to be replaced by Coral Calero, from his own research team. The topics of each session were, respectively: (1) Quantitative approaches using UML, (2) Components and Run-time, (3) Inter and intra class connectivity evaluation and (4) Estimation. On each session three presentations took place, except for the last session where, unfortunately, none of the authors of a position paper originating from the Indian Institute of Technology (Ram, Reddy and Rajasree) did show up. Each presentation, plus corresponding discussion, took around 30 minutes. Those presentations were based on submitted position papers, that went through a revision process conducted by each of the corresponding session chairs. In the final 30-minute wrap-up session, a collective effort was performed to draw some conclusions and to identify future trends for this workshop. Due to the number of participants, active participation

In the next four sections we will present the abstract of each of the position papers presented in each of the workshop sessions, adapted from the one submitted by the corresponding authors. Next, a view of each paper contribution, produced by an independent evaluator (the session chair) is put forward. Finally, a summary of the comments arisen from the consequent discussion among the workshop participants, collected by each session chair, is included. After each position paper title, the list of its co-authors is indicated and the presenter is identified. A table with detailed information about each workshop participant is included in the end of this report.

3 Official Web Site

The call for papers, plus the full text of all accepted position papers are available at: <http://ctp.di.fct.unl.pt/QUASAR/QAOOSE2003/>

4 Session 1 – UML and Product Lines (Chair: Houari Sahraoui)

An OCL-Based Formalization of the MOOSE Metric Suite
Aline Lúcia Baroni, Fernando Brito e Abreu (presenter)

Authors abstract: Most design metrics proposed in the literature are ill-defined, mainly due to its informality. This leads to misinterpretation, namely while producing tools to automate their collection and hampers widespread comparison of obtained metrics values, required to achieve an understanding of their evaluative and predictive

² Participation in ECOOP workshops is usually guaranteed through the submission of a position paper.

power. Besides, since those informal metrics are often not defined upon a design metamodel, their application becomes inadequate in the realm of current industry standard practices for software design. So, in spite of a growing number of empirical research studies, design metrics usage is still not a widespread practice in the software industry.

The well-known MOOSE metrics set, from Chidamber and Kemerer is formalized in this paper, showing, among other things, how the formalization process can help to identify metrics that are unsuitable at design-time. The adopted formal approach to metrics definition was proposed earlier by the QUASAR research group, and uses OCL expressions upon the UML metamodel. Besides solving the above-mentioned ambiguity problems, this approach allows using the metrics definition directly in automating the metrics collection process.

Independent view: The goal of this position paper is to point out the problem of interpretation of existing metrics. Indeed, for most of the existing object-oriented metrics, their definitions are ambiguous and their collection will depend on the target programming language. To circumvent this problem, the authors propose an approach for formalizing object-oriented metrics. This approach is based on the constraint language OCL. It was implemented as a library (FLAME) and tested on a set of metrics (MOOSE metrics).

In its current version, the approach is defined and applied only on structural design metrics (those extracted from UML class diagrams). The authors are currently investigating the possibility of extending it to consider metrics that deal with the behavioral aspects.

Comments summary: Many aspects of this proposal were discussed. First, some participants were interested by the applicability of the approach on dynamic and behavioral diagrams of UML and in particular with the version 2.0. The authors were aware about this aspect and said that is an ongoing work. The second aspect concerned the usefulness of the approach on the particular case of reverse engineering and language semantics. The authors comment that the main goal of the approach is to work on UML diagrams (forward engineering) and in the case of reverse engineering (metric collection from source code) some problems remain unsolved.

Measuring OCL Expressions: a "Tracing"-Based Approach

Luis Reynoso, Marcela Genero, Mario Piattini
Presented by Coral Calero (non-author)

Authors abstract: Since class diagrams constitute the backbone of object-oriented (OO) software development, many metrics were proposed to measure internal quality attributes such as the structural complexity, coupling, size, etc. However, none of the proposed metrics take into account the added complexity when class diagrams are complemented by Object Constraint Language (OCL) expressions. OCL expressions improve class diagrams enhancing their semantic properties, adding precision, improving their documentation and comprehensibility. The importance of OCL and the lack of defined metrics for OCL expressions motivated us to propose a set of

metrics for the structural properties of OCL expressions. The first set of metrics proposed by the authors considers only those OCL concepts related to a "tracing" technique. The authors believe that the "tracing" technique affects the cognitive complexity and, by consequence, the understandability of OCL expressions and the maintenance of a UML class diagram. The main goal of this paper is to show the definition of a set of metrics for structural properties of OCL expressions in a methodological way. The theoretical validation of these metrics according to a property-based framework proposed by Briand et. al is also presented.

Independent view: The objective of this position paper is to discuss the importance of measuring OCL expressions as part of UML design, which is not done yet in the literature and practice. They propose a set of metrics for structural properties of OCL expressions. The authors describe a method that can be used for defining and validating formally and empirically the metrics proposed. Those metrics consider the OCL concepts related to a "tracing" technique which, in the authors' view, affects the cognitive complexity, and by consequence the understandability of OCL expressions and the maintenance of a UML class diagram. The proposed metrics are then theoretically validated using a property-based approach.

Comments summary: The focus of the discussion was on the described method rather than the metrics for OCL themselves. The role of some tasks of this framework (empirical validation, acceptance, etc.) were explained and motivated by the authors.

Quality Modelling for Software Product Lines

Adam Trendowicz (presenter), Teade Punter

Authors abstract. In today's embedded software systems development, non-functional requirements (e.g., dependability, maintainability) become more and more important. Simultaneously the increasing pressure for developing software in shorter time and at a lower cost pushes software industry towards product lines solutions. To support product lines for high quality embedded software, quality models are required. In this paper it is investigated to which extent existing quality modeling approaches facilitate high quality software product lines. First, several requirements for an appropriate quality model are defined. Then, those requirements are used to review the existing quality modeling approaches. The authors conclude from the review that no single quality model fulfils all of the identified requirements. However, several approaches contain valuable characteristics. Based upon those characteristics, the Prometheus approach is proposed. Prometheus is a goal-oriented method that integrates qualitative and quantitative approaches to quality control. The method starts quality modeling early in the software lifecycle and is reusable across product lines.

Independent view: In this position paper, the authors consider the problem of defining quality models for product line solutions. They discuss the possible extents to the existing quality modeling approaches that can facilitate the production of high quality software product lines. To this end, they define the requirements that a quality model must satisfy and use those requirements to review the existing quality modeling approaches. The results of this review show that no single quality model

fulfills all of the requirements. To solve this problem, the authors propose a goal-oriented approach called Prometheus.

Comments summary: One of the characteristics that quality model should have for product lines is the integration of the context in the evaluation. This point was particularly discussed during the workshop. The authors commented that the usage of Bayesian belief networks (BBN) can help considering this characteristic. However, due to confidentiality reasons, it was not possible to see a concrete example of this. Another part of the discussion was dedicated to the problem of the definition of the BBN parameters (conditional probabilities). The authors consider that domain experts should do this. Some participants argued that this is not obvious and an interesting direction to investigate is automatic learning.

5 Session 2 – Components and Run-Time (Chair: Geert Poels)

A Survey on the Quality Information Provided by Software Component Vendors *Manuel F. Bertoa, José M. Troya, Antonio Vallecillo (presenter)*

Authors abstract. The last decade marked the first real attempt to turn software development into engineering through the concepts of Component-Based Software Development (CBSD) and Commercial Off-The-Shelf (COTS) components. The idea is to create high-quality parts and join them together to form a working system. One of the most critical processes in CBSD is the selection of the COTS components from a repository that meet the user requirements.

Current approaches try to propose appropriate quality models for the effective assessment of such COTS components. These proposals define quality characteristics, attributes and metrics, which are specific to the particular nature of COTS components and CBSD. However, the authors have found that the information required to evaluate components using those quality models and metrics is not available in the existing commercial software repositories. In this paper a survey carried out on the most popular COTS component vendor sites is presented, trying to evaluate how much of the information required to assess COTS components is actually available. The authors' goal was to estimate the current gap between the "required" and the "provided" information, since there is no point in defining theoretical measures for COTS components if the data they rest upon is not available. Analyzing this gap is the first step towards successfully bridging it, by both refining the component quality models so their metrics are more realistic, and by improving the information currently provided by software component vendors.

Independent view: In this paper, the authors analyze the gap between the information provided by component vendors and the information that is required to evaluate component quality as part of the COTS component selection process in component-based software development (CBSD). A sample of 164 components from ComponentSource, today's major component marketplace, was used to examine the quality information that is typically offered by component vendors through their web-based systems. The information found was used to assess which of the attributes of

the author's previously developed COTS Component Quality Model (COTS-QM), presented in the QAOOSE'02 workshop, could be measured.

The results show that, based on the information provided, less than half of the attributes in COTS-QM can be used to evaluate quality. In particular, there is almost no information for the quality attributes that have to be measured at run-time. Moreover, the information found was mostly in a raw form (e.g. help-files, manuals, demos, UML diagrams) and required further processing before it could be used in the quality model. Based on their study, the authors recommend that component vendors and acquirers (and quality theoreticians) agree on the type and form (e.g. tagged values like "<metric, value>") of the quality information that is provided. In this context the authors argue for the development of simple, but realistic quality models. They also recommend independent quality assessments and the use of web services for the retrieval of the requested quality information.

Comments summary: The discussion of the paper focused on the quality information that should be minimally available at the vendor's site and on the chances of success regarding the agreement that must be reached with the component vendors. The authors comment that it is still too early to decide on the quality information that must be provided as the proposed quality metrics for components are not validated yet. They do not expect an agreement with vendors in the short term. However, analogous to what happened in the hardware industry, with a push from the buyers, it can be done.

Toward a Definition of Run-Time Object-Oriented Metrics

Aine Mitchell (presenter), James F. Power

Authors abstract. This position paper outlines a programme of research based on the quantification of run-time elements of Java programs. In particular, the authors adapt two common object-oriented metrics, coupling and cohesion, so that they can be applied at run-time. They demonstrate some preliminary results of the analysis they performed on programs from the SPEC JVM98 benchmark suite.

Independent view: Building on the object-oriented metrics definition work of Chidamber and Kemerer, the authors of this paper propose two new coupling metrics (Dynamic CBO for a class, Degree of Dynamic Coupling) and two new cohesion metrics (Simple Dynamic LCOM, Dynamic Call-Weighted LCOM) to measure the external, respectively internal complexity of object-oriented software at run-time. They argue that these metrics deal with the behavioral aspects of a software system, measuring program complexity as actually observed during program execution, whereas the extensive research in the area of object-oriented metrics has focused almost exclusively on metrics for evaluating the structural aspects of a system. As such, these static metrics only quantify what may happen if a program is executed. Several examples were given where the coupling and cohesion of the software at run-time was different from what has been predicted based on a static analysis of the software code. Especially in cases of state-based behavior, dynamic binding, and unequal frequencies of method invocation or instance variables access, run-time metrics provide information of software complexity, not obtained through the classic

(static) metrics. The authors demonstrated their claims by analyzing the complexity of programs from two benchmark suites for the Java programming language. They showed graphically that the run-time profiles of these programs exhibit different levels of coupling and cohesion than the values resulting from calculating the Chidamber and Kemerer CBO and LCOM metrics.

Comments summary: Most of the paper discussion related to the possible use of the information provided by the proposed run-time metrics. Apart from the detection of certain anomalies in the code and possible implications for quantifying the effectiveness of software testing strategies, the authors plan to investigate by means of correlation studies the impact of run-time complexity on external quality attributes. It was also observed that caution must be exercised when interpreting the results of a dynamic analysis because of the dependency on the scenarios and input data that are used. Apart from defining run-time metrics, the research of the authors is therefore aimed at defining a methodology for the sound application of the metrics. They also plan to use their metrics on other than benchmark Java programs, as real-world applications will certainly show different behavior.

CQM: A Software Component Metric Classification Model

Joaquina Martín-Albo (presenter), Manuel F. Bertoa, Coral Calero, Antonio Vallecillo, Alejandra Cechich, Mario Piattini

Authors abstract. In the last few years component-based software development (CBSD) has been imposed as a new paradigm in software systems construction. CBSD is based on the idea that software systems can be constructed by selecting and integrating appropriate components, which have already been developed, and then assembling them to obtain the functionality desired in the final application. Multiple authors have proposed metrics to quantify several components characteristics in order to help in its selection. Nevertheless, rather than helping developers, such proposals often provoke more confusion due to the fact that they do not systematically take into account different aspects of the components.

Trying to achieve clarity in this line, the authors have developed the CQM model (Component Quality Model), whose first aim is to propose a classification of the defined metrics for software components. The model will also be suited both for the evaluation of a component or a component system. Finally, it is necessary to indicate that this is the first version of the model, and that it requires refinement by means of its use and discussion in different fora.

Independent view: This paper reports upon a joint effort of researchers from several universities into the quality of component-based software development (CBSD). The authors present the Component Quality Model (CQM), a comprehensive quality evaluation framework for components and component-based systems. The CQM model proposes four dimensions that must be considered in the quality evaluation: (i) the quality characteristic (distinguishing between internal quality, external quality and quality in use properties), (ii) the granularity (system or component) and visibility (internal or external) dimension, (iii) the lifecycle process involved and (iv) the stakeholder involved. Due to its comprehensive nature, the model integrates, in some

of its dimensions, previously proposed quality models for components, like the COTS-QM model of Bertoa and Vallecillo, thereby linking with the first 'quality in CBSD'-related paper in this session.

A first use of the model is to classify the metrics that have been proposed in the literature for the characterization, evaluation and selection of components. The authors demonstrate this idea by classifying a number of usability metrics along the four dimensions. They also point out that the model presented is only a first version, which needs to be further refined. The new version of the model does, for instance, not consider the stakeholder, as this dimension does not seem to be orthogonal to the life cycle process.

Comments summary: During the paper discussion the intended use of the CQM model was clarified. It was suggested that the metrics classification could be used to identify both neglected spaces (and hence initiate new metric definition work) and conflicts, like when researchers propose the same metrics, but with different names. It was further stressed that the ultimate goal of CQM is to evaluate components and component-based systems, giving stakeholders an instrument to select the appropriate quality metrics. Regarding a workshop participant's question on the validity of classified metrics in particular contexts, it was answered that for the usability metrics classified, their validity was not really questioned. On the other hand, the authors point out that besides the classification along the four dimensions, other metric characteristics need to be identified, including their theoretical and empirical validation, their objective or subjective nature, and whether there is automated support for their collection and calculation.

6 Session 3 – Inter and Intra Class Connectivity Evaluation (Chair: Fernando Brito e Abreu)

A New Class Cohesion Criterion: An Empirical Study on Several Systems

Linda Badri, Mourad Badri (presenter)

Authors abstract: Class cohesion is considered one important object-oriented software attribute. Cohesion refers to the degree of the relatedness of the members in a class. High cohesion is a desirable property of classes. Several metrics have been proposed in the literature in order to measure class cohesion in object-oriented systems. They capture class cohesion in terms of connections among members within a class. The major existing class cohesion metrics are essentially based on instance variables usage criteria. It is a special and a restricted way of capturing class cohesion. Most of these metrics have been experimented and widely discussed in the literature. They do not take into account some characteristics of classes as stated in many papers. The authors believe that class cohesion metrics should not exclusively be based on common instance variables usage. They present, in this position paper, a new criterion, which focuses on interactions between class methods and propose a new class cohesion metric. The authors have developed a cohesion measurement tool for Java programs and performed a case study on several systems. The selected test

systems vary in size and domain. The obtained results demonstrate that their new class cohesion metric captures several pairs of related methods, which are not captured by the existing cohesion metrics.

Independent view: The authors argue that the traditional view of class cohesiveness, based upon measuring the usage of attributes by operations, is just a partial view of that property. Therefore, they explore another dimension of class cohesiveness, based upon the interaction among class operations or, in other words, dependent on the internal invocation of operations. They call this property the functional cohesion. The authors then proposed two metrics for this property, DC_o , which is the percentage of public operation pairs that are directly related and DC_i , which is the percentage of public operation pairs that are indirectly related. That relatedness is evaluated using only attribute cohesiveness or attribute and operation cohesiveness together.

An empirical study using a cohesion measurement tool for Java programs is also reported. That tool, built by the authors on top of a well-known parser (antlr), allows computing several published cohesion metrics, plus the ones defined in this paper. A data set of six free downloadable Java systems, summing up more than 2000 classes, was used. The experiment concluded that there is statistically significant difference between the cohesiveness metrics calculated with attribute and operation together, and the ones that consider attributes usage only.

Comments summary: A participant commented that it is not unexpected that adding a factor to an existing cohesion metric, would naturally increase the number of connected class pairs. Another participant confirmed with the presenting author that the proposed metrics are only static, that is, obtained from the design structure and not from running hypothetical scenarios. It was commented that the proposed metrics could also be applied to components, in a white box approach. Finally, the need for empirical validation of the proposed metrics usefulness was arisen.

Towards a Validation Process for Measuring Coupling: Integrating Axiomatic and Empirical Approaches

Miguel Lopez, Valérie Paulus (presenter), Naji Habra

Authors abstract: The validity of the measures used in software engineering is a critical matter about which no consensus has yet emerged, although it has prompted hard discussion. There is a need for unambiguous definitions of the mathematical properties that characterize the major measurement concepts. Such a mathematical framework could help to generate consensus among the software engineering community. The goal of this paper is to provide a formal validation process for software measurement. It presents a global measurement framework that integrates theoretical and empirical validation processes based on measurement theory.

The concept underlying the framework is to formalize some properties of the measure to be analyzed, and then to verify the conformity of these properties to the measure by means of formal experimentation.

This validation process determines a contextual validity (scope) defined by the set of factors or validity conditions that impact the validity of the measure. The paper

develops a case study that, under specified conditions, validates the Coupling Between Objects (CBO) as a measure of coupling.

Independent view: This paper deals with the recurring issue of validating software measures. In particular, it aims at presenting a process that combines theoretical (measurement theory based) and empirical validation. To illustrate the process, a case study is presented, that tries to validate the C&K Coupling Between Object Classes metric as a coupling metric. The authors use UML class diagrams to review the different coupling types among classes. They restrict their coupling scope to attribute and operation dependency.

The kind of validity the authors aim to deal with has to do with suitability, rather than usefulness. For instance, given a metric, they want to validate its suitability as a coupling metric, rather than validating it as a useful predictor or indicator of a given external characteristic, such as maintainability. The proposed validation process is composed of eight steps. It involves a set of OO design experts, responsible for (i) defining a required set of axioms mathematically and (ii) for sorting a set of fragments of class diagrams supposed to illustrate, as exhaustively as possible, alternative coupling topologies (called coupling paradigms in the paper). The former requirement is left unexplained since the authors use a set of well-known published axioms.

The two last steps of the proposed process are the ones where the validation is actually performed, being the first of them related to the empirical world and the used measurement instrument, and the second about the preservation of the relationship between the empirical and the numerical worlds.

Although the authors raise doubts on the problem of how to gain confidence on experts judgement and also on the representativeness of the coupling paradigms, the result of their validation case study allows to invalidate the Coupling Between Object Classes metric as a coupling metric.

Comments summary: The two authors present were questioned on the non-appropriateness of the Spearman correlation coefficient for order scales. One participant also mentioned the need for cross validation of the case study results. Another participant also questioned the "blind" adoption of the Briand and Morasca axioms and mentioned, for instance, how the axiom of non-negativity could be undesirable. The discussion motivated by this paper ended around the always-pervasive problem of obtaining representative sets for validation purposes.

Evolution of Cyclomatic Complexity in Object Oriented Software

Rajesh Vasa (presenter), Jean-Guy Schneider

Authors abstract: It is a generally accepted fact that software systems are constructed and gradually refined over a period of time. During this time, code is written and modified until stable releases of the system emerge. Many researchers have studied systems over a longer period of time in order to understand how they change and evolve. Despite these efforts, the authors feel that there is still a lack of a precise understanding how various properties of software change over time, in particular in the area of object-oriented systems. Such an understanding is of great importance if one wants to come up with techniques to provide feedback on the

evolution of quality and predictions about further evolution of software systems. Historically, collection of sufficient data to build useful models was not practical as source code and build histories were not freely available. The authors argue that by focusing their attention towards Open source software repositories, they will have a better hope building predictive models to help developers and managers. In this paper, the authors report on an exploratory study analyzing Open source object oriented software projects and present a first predictive model based on this analysis.

Independent view: The authors claim that few works were published on the evolution of OO systems, despite the availability of evolution data stored in open source repositories. The paper includes data from 5 Java projects throughout time (data ranging from 7 to 10 different versions each). Using a public domain metrics collection tool (JavaNCSS), the authors analyzed and commented the evolution of NCSLOC (non-commented source lines of code), confirming Lehman's "first law of software evolution". Then they analyzed the cyclomatic complexity of operations for which an implementation was available. The operations were grouped in five categories ranked with cyclomatic complexity intervals. Observation of results led to the confirmation of Lehman's "fifth law of software evolution" for object-oriented software.

The authors look after producing a predictive model of cyclomatic complexity based on the observed data, that could warn producers of any abnormal evolution. However, that model is left for future work in this paper.

Comments summary: A few questions emerged after the paper presentation. One was on the distribution of size versus complexity. The co-author present claimed that one had no impact on the other. He also mentioned that they are currently working on evaluating the "regularity" of graphs displaying the complexity categories throughout time. A participant suggested that the evolution of the shapes of the curves of cyclomatic complexity should be compared.

A question was also risen by one participant on the need for such a evolution study: if a large system evolves through non-disruptive steps, where consecutive versions differ very few, why bother measuring the difference? The author replied that the large number of plain selectors and modifiers contributed a lot to that regularity. If those operations were not considered, than the distinctions would be much more noticeable.

7 Session 4 – Estimation (Chair: Coral Calero)

Analogy-Based Software Quality Prediction

David Grosser, Houari A. Sahraoui (presenter), Petko Valtchev

Authors abstract: Predicting the stability of object-oriented systems is an important and challenging task. Classical approaches to quality prediction perform some form of inductive inference starting from datasets of software items with known quality factor values and looking for typical features that discriminate the items regarding the quality factor. However, most of the effective methods for predictive model

construction are based on the implicit hypothesis that the available sample representative, which is rather strong. The approach the authors propose implements a similarity-based comparison principle. In it, the quality factor (stability) of a software item is estimated from the recorded stability of a set of other items that been recognized as the most similar to that item among a larger set of items stored in a database. This approach is evaluated using the successive versions of the JDK 1

Independent view: The authors propose an approach for predicting the stability of object-oriented systems. This approach implements a similarity-based comparison principle: the stability of a given software item is estimated from the recorded stability of a set of other items that have been recognized as the most similar to that item among a larger set of items stored in a database. The authors use CBR (Case Based Reasoning) as a suitable approach to the software stability prediction problem because they have carried out some experiments that confirmed the usefulness of CBR for tasks where few theoretical knowledge is available. Finally, the authors state that this technique can also be applied to other quality factors prediction.

Comments summary: During the questions, authors were asked about the dependency of this estimation technique on the metrics selected. They argued since this is the problem with estimation in general, they tried to apply the technique under the hypothesis of metrics set independence. Another question was why prefer to use prediction instead of pair correlation. The presenter replied that correlation is a good technique for knowing if a variable has an impact in another but not for estimation.

Definition and Validation of a COSMIC-FFP Functional Size Measure for Object-Oriented Systems

Geert Poels (presenter)

Author abstract: COSMIC Full Function Points is an ISO approved functional measurement method for modeling and sizing software systems, based on functional user requirements. Recently, a number of mappings have been proposed from the COSMIC-FFP meta-model onto the concepts used in UML and other modeling approaches. Given the many problems with COSMIC-FFP's predecessor, Function Points Analysis, it is necessary to establish confidence in the validity of COSMIC-FFP as a functional size measure for OO systems, before widespread diffusion in quantitative OO software engineering practice. In this paper the author presents an attempt at validating COSMIC-FFP using distance-based software measurement. This theoretical validation approach for software measurement is firmly grounded in Measurement Theory and has been applied before in the validation of OO software measures.

Independent view: The author presented the formal validation of the COSMIC-FFP using the distance-based software measurement, an approach based on measurement theory that has been previously applied to OO software measurement. COSMIC Full Function Points is an ISO approved functional size measurement method for modeling and sizing software systems based on their functional requirements.

The COSMIC-FFP metamodel of functional user requirements (FUR) for modeling and sizing a software system includes a set of modelling concepts. This metamodel is used to map the FUR that are captured in the operational requirements engineering or software engineering artifacts into a uniform COSMIC-FFP FUR model, which is subsequently measured. The COSMIC-FFP measurement function takes a piece of software as its argument and returns a value, representing the functional size of that piece of software. The mapping rules to generate a COSMIC-FFP FUR model are designed to be applicable to a broad range of artifacts, such as high-level software specifications that are documented in UML class diagrams and use case diagrams, UML, Real-time Object-Oriented Modelling methods, etc.

The final objective of the author was to prove if the metric could be considered valid with respect to the measurement theory. A theoretically validated measure means that we can evaluate the construct validity of the measure when used as a measurement instrument in an empirical study. The attempt to validate the COSMIC-FFP measurement function, as a ratio scale level functional size measure for OO software systems, was not successful. However, the distance-based software measurement demonstrates that the view of functional size that is inherent in the COSMIC-FFP metamodel allows measuring this property at the ratio scale level.

Comments summary: One workshop participant questioned the need to establish a mapping between the concepts in the COSMIC-FFP metamodel and the concepts used in UML and other OO modeling approaches. After all, COSMIC-FFP is designed to be independent of modeling paradigms. The author replied that although the COSMIC-FFP metamodel is generic and highly abstract, the application of COSMIC-FFP in practice requires more specific and detailed rules that must necessarily be tailored to the modeling paradigm used. Another question related to the measurement theoretic structures on which distance-based software measurement is based. It was answered that this approach to theoretical metrics validation belongs to an advanced type of measurement that is known as segmentally additive proximity measurement. The author presented more details about this approach at the QAOOSE'99 workshop.

8 Conclusions

In the final wrap-up session the name of the QAOOSE workshop series was questioned. There was a suggestion to broaden up the scope of the workshop to quantitative approaches for other than object-oriented modeling, specification and programming methodologies and technologies. In particular, workshop participants agreed that quality and process related research for component, web services and agent-based systems should fit into the workshop. To some extent, this year's edition has already paved the way to this, as it can be confirmed by reading the call for papers. It was also suggested not to change the name of the workshop series, since the underlying technology for these newer trends in software engineering is mostly still object orientation. Nevertheless, the organizers expressed the will to extensively review the list of workshop topics, namely by considering more recent development paradigms, thus opening the workshop participation to a broader audience. This edition of the workshop was definitely one of the most successful ones, mainly due to an active participation of all attendees. The support and suggestions raised by

the new ideas presented by some participants and the feeling of interact interest of all the people in the workshop has resulted on prospects for colla research among different participants.

Let us finish this report with a strategy remark. Looking back to the particip of this edition, plus those of previous ones, we cannot avoid the observation vast majority of participants originated from academia and research cen latter somehow in-between universities and industry. In other words, QAO failed to attract people from "pure" industry, a situation that the Q organizers feel is undesirable, for bridging the gap between object quantitative software engineering research and practice. This situation aris the fact that this workshop is an embedded event of ECOOP, which, in tu event dominated by and mainly targeted at academic researchers. Therefore be a good idea to collocate QAOOSE with some other event where a participation is not so much dominant.

9 Participants' Affiliation and Contacts

The following table includes detailed information about all participants workshop, with their corresponding roles (O - Organizer; A - Author or co-a - Present in workshop).

Name	Surname	Affiliation	Country	Email
Miguel	Lopez	CETIC Falcués de Namur	Belgium	malm@cetic.be mlo@info.fundp.ac.be
Valérie	Paulus	CETIC Falcués de Namur	Belgium	vp@cetic.be vpa@info.fundp.ac.be
Naji	Habra	CETIC / Falcués de Namur	Belgium	nha@info.fundp.ac.be
Adam	Trendowicz	Fraunhofer IESE	Germany	trend@iese.fhg.de
Geert	Poels	Ghent University Katholieke Universiteit Leuven	Belgium	geert.poels@ugent.be geert.poels@econ.kuleuven c.be
P. Jithendra	Kumar Reddy	Indian Institute of Technology	India	jithendra@cs.iitm.ernet.in
M. S.	Rajasree	Indian Institute of Technology	India	rajasree@cs.iitm.ernet.in
D. Janaki	Ram	Indian Institute of Technology	India	djram@cs.iitm.ernet.in
James	F. Power	National University of Ireland	Ireland	jpowers@cs.may.ie
Aine	Mitchel	National University of Ireland	Ireland	ainem@cs.may.ie
Jean-Guy	Schneider	Swinburne Univers. of Technology	Australia	jschneider@swin.edu.au
Rajesh	Vasa	Swinburne Univers. of Technology	Australia	rvasa@swin.edu.au
Linda	Badri	Univ. du Québec à Trois- Rivières	Canada	linda_badri@uqtr.ca
Mourad	Badri	Univ. du Québec à Trois- Rivières	Canada	mourad_badri@uqtr.ca
Alejandra	Cechich	Univ. Nacional del Comahue	Argentina	acechich@uncoma.edu.ar

Luis	Reynoso	Univ. Nacional del Comahue	Argentina	lreynoso@uncoma.edu.ar	A
Coral	Calero	Universidad de Castilla-La Mancha	Spain	coral.calero@uclm.es	AP
Marcela	Genero	Universidad de Castilla-La Mancha	Spain	marcela.genero@uclm.es	A
Joaquina	Martín-Albo	Universidad de Castilla-La Mancha	Spain	jmartin@proyectos.inf-cr.uclm.es	AP
Mario	Piattini	Universidad de Castilla-La Mancha	Spain	mario.piattini@uclm.es	OA
Manuel	F. Bertoa	Universidad de Malaga	Spain	bertoa@lcc.uma.es	AP
José	M. Troya	Universidad de Malaga	Spain	troya@lcc.uma.es	A
Antonio	Vallecillo	Universidad de Malaga	Spain	av@lcc.uma.es	AP
Aline	Baroni	Universidade Nova de Lisboa	Portugal	baroni@di.fct.unl.pt	A
Fernando	Brito e Abreu	Universidade Nova de Lisboa INESC-ID	Portugal	fba@di.fct.unl.pt fba@inesc.pt	OAP
Hervé	Paulino	Universidade Nova de Lisboa	Portugal	herve@di.fct.unl.pt	P
David	Grosser	Université de Montréal	Canada	grosserd@iro.umontreal.ca	A
Houari	Sahraoui	Université de Montréal	Canada	sahraouh@iro.umontreal.ca	OAP
Petko	Valtchev	Université de Montréal	Canada	valtchev@iro.umontreal.ca	A
Hans	Stenten	University of Antwerp	Belgium	hans.stenten@ua.ac.be	P
Filip	Van Rysselberghe	University of Antwerp	Belgium	filip.vanrysselberghe@ua.ac.be	P
Soren	Gaardbo Jensen	University of Copenhagen	Denmark	gaardbo@dkv.dk	P
Katharina	Mehner	University of Paderborn	Germany	mehner@upd.de	P

10 Related Workshops

This workshop is a direct continuation of the QA00SE series of workshops, held at previous ECOOP conferences:

Workshop Name	Conference/ Place	Organizers
QA00SE2002 http://alarcos.inf-cr.uclm.es/qa00se2002	ECOOP'2002 Malaga, Spain	Mario Piattini, F. Brito e Abreu, Geert Poels, Houari Sahraoui
QA00SE2001 http://www.iro.umontreal.ca/~sahraouh/qa00se01/	ECOOP'2001 Budapest, Hungary	F. Brito e Abreu, Brian Henderson-Sellers, Mario Piattini, Geert Poels, Houari Sahraoui
QA00SE2000 http://ecoop2000.unice.fr/Program/Technical/Workshops/w10.html	ECOOP'2000 Cannes, France	F. Brito e Abreu, Geert Poels, Houari Sahraoui, Horst Zuse
QA00SE99 http://ecoop99.di.fc.ul.pt/techprogramme/w20.html	ECOOP99 Lisbon, Portugal	F. Brito e Abreu, Walcelio Melo, Houari A. Sahraoui, Horst Zuse
OO Product Metrics for Software Quality Assessment http://www.crim.ca/~hsahraou/ooqm.html	ECOOP98 Brussels, Belgium	Houari A. Sahraoui, Sandro Morasca, Walcelio Melo
Quantitative Methods for OO Systems Development http://ctf.di.fct.unl.pt/QUASAR/ECOOP95	ECOOP95 Aarhus, Denmark	Horst Zuse, Brian Henderson-Sellers, F. Brito e Abreu