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.)

More recently, several color-cover sublines have been added featuring, beyond a collection of papers, various added-value components; these sublines include

- tutorials (textbook-like monographs or collections of lectures given at advanced courses)
- state-of-the-art surveys (offering complete and mediated coverage of a topic)
- hot topics (introducing emergent topics to the broader community)

In parallel to the printed book, each new volume is published electronically in LNCS Online.

Detailed information on LNCS can be found at *http://www.springeronline.com*

Proposals for publication should be sent to

LNCS Editorial, Tiergartenstr. 17, 69121 Heidelberg, Germany

E-mail: lncs@springer-sbm.com

ISSN 0302-9743

ISBN 3-540-29395-7

9 783540 293958

Lecture Notes in Computer Science

LNCS LNAI LNBI

› springeronline.com

Akoka et al. (Eds.)

LNCS 3770

Perspectives in Conceptual Modeling

ER 2005 Workshops

LNCS 3770

Jacky Akoka et al. (Eds.)

Perspectives in Conceptual Modeling

ER 2005 Workshops AOIS, BP-UML CoMoGIS, eCOMO, and QoIS Klagenfurt, Austria, October 2005, Proceedings

Springer

Lecture Notes in Computer Science 3770

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

Jacky Akoka Stephen W. Liddle
Il-Yeol Song Michela Bertolotto
Isabelle Comyn-Wattiau Samira Si-Saïd Cherfi
Willem-Jan van den Heuvel Bernhard Thalheim
Manuel Kolp Paolo Bresciani Juan Trujillo
Christian Kop Heinrich C. Mayr (Eds.)

Perspectives in Conceptual Modeling

ER 2005 Workshops AOIS, BP-UML
CoMoGIS, eCOMO, and QoIS
Klagenfurt, Austria, October 24-28, 2005
Proceedings

Springer

members of the organization team, who gave their best to make ER 2005 an unforgettable event. Last but not least we thank our sponsors and supporters, in particular the University of Klagenfurt, the Governor of Carinthia, and the Mayor of Klagenfurt, who helped us to organize a high-level event at comparably low cost.

The workshop and tutorial co-chairs also express our deep gratitude and respect for the ER 2005 General Chair, Heinrich C. Mayr, whose leadership and organization were outstanding. The high-quality program is a reflection of the countless hours he spent working so hard for all of us.

October 2005

Jacky Akoka
Stephen W. Liddle
Il-Yeol Song

ER 2005 Conference Organization

Honorary Conference Chair

Peter P.S. Chen — Lousiana State University, USA

General Conference Chair

Heinrich C. Mayr — Alpen-Adria University of Klagenfurt, Austria

Scientific Program Co-chairs

Lois M.L. Delcambre	Portland State University, Portland, USA
John Mylopoulos	University of Toronto, Canada
Oscar Pastor López	Universitat Politècnica de València, Spain

Workshop and Tutorial Co-chairs

Jacky Akoka	CEDRIC – CNAM, France/Institut National des Télécommunications, Evry, France
Stephen W. Liddle	Brigham Young University, Provo, USA
Il-Yeol Song	Drexel University, Philadelphia, USA

Panel Chair

Wolfgang Hesse — Philipps-Universität Marburg, Germany

Demos and Posters Chair

Tatjana Welzer — University of Maribor, Slovenia

Industrial Program Chair

Andreas Schabus — Microsoft Austria, Vienna, Austria

ER Steering Committee Liaison Manager

Veda Storey	Georgia State University, USA

Joint Conferences Steering Committee Co-chairs

Ulrich Frank	University of Essen-Duisburg, Germany
Jörg Desel	Catholic University Eichstätt-Ingolstadt, Germany

Organization and Local Arrangements

Markus Adam	Peter Jelitsch	Christine Seger
Stefan Ellersdorfer	Christian Kop	Claudia Steinberger
Günther Fliedl	Heinrich C. Mayr	
Robert Grascher	Alexander Salbrechter	

ER 2005 Workshops Organization

ER 2005 Workshop and Tutorial Chairs

Jacky Akoka	CEDRIC – CNAM, France/Institut National des Télécommunications, Evry, France
Stephen W. Liddle	Brigham Young University, Provo, USA
Il-Yeol Song	Drexel University, Philadelphia, USA

BP-UML 2005 Program Chair

Juan Trujillo	Universidad de Alicante, Spain

AOIS 2005 Program Chairs

Manuel Kolp	Université Catholique de Louvain, Belgium
Paolo Bresciani	ITC-irst, Italy

CoMoGIS 2005 Program Chair

Michela Bertolotto	University College Dublin, Ireland

eCOMO 2005 Program Chairs

Willem-Jan van den Heuvel	Tilburg University, The Netherlands
Bernhard Thalheim	Christian-Albrechts-Universität Kiel, Germany

QoIS 2005 Program Chairs

Isabelle Comyn-Wattiau	CEDRIC – CNAM, France/ESSEC, France
Samira Si-Saïd Cherfi	CEDRIC – CNAM, France

BP-UML 2005 Program Committee

Doo-Hwan Bae	KAIST, South Korea
Michael Blaha	OMT Associates Inc., USA
Cristina Cachero	Universidad de Alicante, Spain
Eduardo Fernandez	Universidad de Castilla-La Mancha, Spain
Paolo Giorgini	Università di Trento, Italy
Jaime Gómez	Universidad de Alicante, Spain
Peter Green	University of Queensland, Australia
Manfred Jeusfeld	Tilburg University, The Netherlands
Ludwik Kuzniarz	Blekinge Institute of Technology, Sweden
Jens Lechtenbörger	University of Münster, Germany
Pericles Loucopoulos	University of Manchester, UK
Kalle Lyytinen	Case Western Reserve University, Cleveland, USA
Hui Ma	Massey University, New Zealand
Antoni Olive	Universitat Politècnica de Catalunya, Spain
Andreas L. Opdahl	University of Bergen, Norway
Jeff Parsons	Memorial University of Newfoundland, Canada
Oscar Pastor López	Universitat Politècnica de València, Spain
Witold Pedrycz	University of Alberta, Canada
Mario Piattini	Universidad de Castilla-La Mancha, Spain
Colette Rolland	Université Paris 1 Panthéon-Sorbonne, France
Matti Rossi	Helsinki School of Economics, Finland
Manuel Serrano	Universidad de Castilla-La Mancha, Spain
Keng Siau	University of Nebraska-Lincoln, USA
Il-Yeol Song	Drexel University, Philadelphia, USA
Bernhard Thalheim	Christian-Albrechts-Universität Kiel, Germany
Ming Tjoa	Vienna University of Technology, Austria
Ambrosio Toval	Universidad de Murcia, Spain
Panos Vassiliadis	University of Ioannina, Greece

BP-UML 2005 External Referees

James Goldman	Drexel University, Philadelphia, USA
Mary E. Jones	Drexel University, Philadelphia, USA
Francisco Lucas Martinez	Universidad de Murcia, Spain
Fernando Molina	Universidad de Murcia, Spain
Joaquin Nicolas	Universidad de Murcia, Spain
Choi Nam Youn	KAIST, South Korea
Faizal Riaz-ud-Din	Massey University, New Zealand

AOIS 2005 Steering Committee

Yves Lespérance	York University, Toronto, Canada
Gerd Wagner	Technische Universiteit Eindhoven, The Netherlands
Eric Yu	University of Toronto, Canada
Paolo Giorgini	Università di Trento, Italy

AOIS 2005 Program Committee

Carole Bernon	Université Paul Sabatier, Toulouse 3, France
Brian Blake	Georgetown University, Washington, DC, USA
Paolo Bresciani	ITC-irst, Italy
Jaelson Castro	Federal University of Pernambuco, Brazil
Luca Cernuzzi	Universidad Católica "Nuestra Señora de la Asunción", Paraguay
Massimo Cossentino	ICAR-CNR, Palermo, Italy
Luiz Cysneiros	York University, Toronto, Canada
John Debenham	University of Technology, Sydney, Australia
Scott DeLoach	Kansas State University, USA
Frank Dignum	University of Utrecht, The Netherlands
Paolo Donzelli	University of Maryland, College Park, USA
Bernard Espinasse	Domaine Universitaire de Saint-Jérôme, France
Stéphane Faulkner	University of Namur, Belgium
Behrouz Homayoun Far	University of Calgary, Canada
Innes A. Ferguson	B2B Machines, USA
Alessandro Garcia	PUC Rio de Janeiro, Brazil
Chiara Ghidini	ITC-irst, Italy
Aditya K. Ghose	University of Wollongong, Australia
Marie-Paule Gleizes	Université Paul Sabatier, Toulouse 3, France
Cesar Gonzalez-Perez	University of Technology, Sydney, Australia
Giancarlo Guizzardi	University of Twente, The Netherlands
Igor Hawryszkiewycz	University of Technology, Sydney, Australia
Brian Henderson-Sellers	University of Technology, Sydney, Australia
Carlos Iglesias	Technical University of Madrid, Spain
Manuel Kolp	Université Catholique de Louvain, Belgium
Daniel E. O'Leary	University of Southern California, USA
C. Li	University of Technology, Sydney, Australia
Graham Low	University of New South Wales, Australia
Carlos Lucena	PUC Rio de Janeiro, Brazil
Philippe Massonet	CETIC, Belgium

Haris Mouratidis	University of East London, UK
Jörg P. Müller	Siemens, Germany
Juan Pavón	Universidad Complutense Madrid, Spain
Omer F. Rana	Cardiff University, UK
Onn Shehory	IBM Haifa Labs, Israel
Nick Szirbik	Technische Universiteit Eindhoven, The Netherlands
Kuldar Taveter	VTT Information Technology, Finland/ University of Melbourne, Australia
Nhu Tran	University of New South Wales, Australia
Viviane Torres da Silva	PUC Rio de Janeiro, Brazil
Michael Winikoff	RMIT, Australia
Carson Woo	University of British Columbia, Vancouver, Canada
Bin Yu	North Carolina State University, USA
Amir Zeid	American University in Cairo, Egypt
Zili Zhang	Deakin University, Australia

AOIS 2005 External Referees

Alessandro Oltramari	LOA-CNR, Italy
Ghassan Beydoun	University of New South Wales, Australia
Renata S.S. Guizzardi	ITC-irst, Italy
Rodrigo de Barros Paes	PUC Rio de Janeiro, Brazil
Aneesh Krishna	University of Wollongong, Australia
Gustavo Robichez de Carvalho	PUC Rio de Janeiro, Brazil

CoMoGIS 2005 Program Committee

Natalia Andrienko	Fraunhofer Institut AIS, Germany
Masatoshi Arikawa	University of Tokyo, Japan
Patrice Boursier	Université de La Rochelle, France/ Open University, Malaysia
Elena Camossi	IMATI-CNR, Genoa, Italy
James Carswell	Dublin Institute of Technology, Ireland
Christophe Claramunt	Naval Academy, France
Maria Luisa Damiani	University of Milan, Italy
Max Egenhofer	University of Maine, USA
Stewart Fotheringham	NCG, Maynooth, Ireland
Andrew Frank	Vienna University of Technology, Austria
Michael Gould	Universitat Jaume I, Spain
Jihong Guan	Wuhan University, China
Bo Huang	University of Calgary, Canada
Zhiyong Huang	National University of Singapore, Singapore
Christian S. Jensen	Aalborg University, Denmark
Menno-Jan Kraak	ITC, Enschede, The Netherlands
Ki-Joune Li	Pusan National University, South Korea
Dieter Pfoser	CTI, Greece
Philippe Rigaux	Université Paris-Sud 11, France
Andrea Rodriguez	University of Concepcion, Chile
Sylvie Servigne	INSA, France
Stefano Spaccapietra	EPFL, Switzerland
George Taylor	University of Glamorgan, UK
Nectaria Tryfona	Talent Information Systems/ Hellenic Open University, Greece
Christelle Vangenot	EPFL, Switzerland
Agnes Voisard	Fraunhofer ISST, Germany/ Freie Universität Berlin, Germany
Nancy Wiegand	University of Wisconsin-Madison, USA
Thomas Windholz	Idaho State University, USA
Stephan Winter	University of Melbourne, Australia
Ilya Zaslavsky	San Diego Supercomputer Center, USA
Shuigeng Zhou	Fudan University, China

eCOMO 2005 Program Committee

Fahim Akhter	Zayed University, United Arab Emirates
Boldur Barbat	Lucian Blaga University, Sibiu, Romania
Boualem Benatallah	University of New South Wales, Australia
Anthony Bloesch	Microsoft Corporation, USA
Antonio di Leva	University of Turin, Italy
Vadim A. Ermolayev	Zaporozhye State Univ., Ukraine
Marcela Genero	Universidad de Castilla-La Mancha, Spain
Martin Glinz	University of Zurich, Switzerland
József Györkös	University of Maribor, Slovenia
Bill Karakostas	UMIST Manchester, UK
Roland Kaschek	Massey University, New Zealand
Christian Kop	Alpen-Adria University of Klagenfurt, Austria
Jacques Kouloumdjian	INSA, Lyon, France
Stephen W. Liddle	Brigham Young University, Provo, USA
Zakaria Maamar	Zayed University, United Arab Emirates
Norbert Mikula	Intel Labs, Hillsboro, USA
Oscar Pastor López	Universitat Politècnica de València, Spain
Barbara Pernici	Politecnico di Milano, Italy
Matti Rossi	Helsinki School of Economics, Finland
Klaus-Dieter Schewe	Massey University, New Zealand
Joachim Schmidt	Technische Universität Hamburg-Harburg, Germany
Michael Schrefl	Johannes-Kepler-Universität Linz, Austria
Daniel Schwabe	PUC Rio de Janeiro, Brazil
Il-Yeol Song	Drexel University, Philadelphia, USA
Eberhard Stickel	Europa-Universität Viadrina, Frankfurt(Oder), Germany
Bernhard Thalheim	Christian-Albrechts-Universität Kiel, Germany
Jos van Hillegersberg	Erasmus University Rotterdam, The Netherlands
Ron Weber	University of Queensland, Australia
Carson Woo	University of British Columbia, Vancouver, Canada
Jian Yang	Tilburg University, The Netherlands

QoIS 2005 Program Committee

Jacky Akoka	CEDRIC – CNAM, France/Institut National des Télécommunications, Evry, France
Laure Berti	IRISA, France
Mokrane Bouzeghoub	Université de Versailles Saint-Quentin-en-Yvelines, France
Tiziana Catarci	Università di Roma "La Sapienza", Italy
Corinne Cauvet	Université Aix-Marseille III, France
Marcela Genero	Universidad de Castilla-La Mancha, Spain
Michael Gertz	University of California, Davis, USA
Markus Helfert	Dublin City University, Ireland
Paul Johannesson	Stockholm University, Sweden
Jacques Le Maitre	Université du Sud Toulon-Var, France
Oscar Pastor López	Universitat Politècnica de València, Spain
Geert Poels	Ghent University, Belgium
Nicolas Praat	ESSEC, France
Jolita Ralyté	University of Geneva, Switzerland
Il-Yeol Song	Drexel University, Philadelphia, USA
Bernhard Thalheim	Christian-Albrechts-Universität Kiel, Germany
Juan Trujillo	Universidad de Alicante, Spain
Dimitri Theodoratos	New Jersey Institute of Technology, USA

QoIS 2005 External Referees

Mary E. Jones	Drexel University, Philadelphia, USA
Ann Maes	Ghent University, Belgium
Luis Reynoso	Universidad Nacional del Comahue, Neuquén, Argentina
Manuel Serrano	Universidad de Castilla-La Mancha, Spain

Organized by

Institute of Business Informatics and Application Systems, Alpen-Adria University of Klagenfurt, Austria

Sponsored by

ER Institute
The Governor of Carinthia
The City Mayor of Klagenfurt

In Cooperation with

GI Gesellschaft für Informatik e.V.
Austrian Computer Society

Table of Contents

First International Workshop on Best Practices of UML (BP-UML 2005)

Experience Reports and new Applications

Model Evaluation and Requirements Modeling

Metamodeling and Model Driven Development

Seventh International Bi-conference Workshop on Agent-Oriented Information Systems (AOIS-2005)

Invited Talk

Positions in Engineering Agent Oriented Systems

Agent Oriented Methodologies and Conceptual Modeling

Agent Communication and Coordination

Second International Workshop on Conceptual Modeling for Geographic Information Systems (CoMoGIS 2005)

Invited Talk

Spatial and Spatio-temporal Data Representation

Spatial Relations

Spatial Queries, Analysis and Data Mining

Data Modeling and Visualisation

Sixth International Workshop on Conceptual Modeling Approaches for e-Business (eCOMO 2005)

First International Workshop on Quality of Information Systems (QoIS 2005)

Information System Models Quality

Quality Driven Processes

Tutorials

Preface to BP-UML 2005

Juan Trujillo

The Unified Modeling Language (UML) has been widely accepted as the standard object-oriented (OO) modeling language for modeling various aspects of software and information systems. The UML is an extensible language, in the sense that it provides mechanisms to introduce new elements for specific domains if necessary, such as web applications, database applications, business modeling, software development processes, data warehouses and so on. Furthermore, the latest approach of the Object Management Group (OMG) surrounding the UML even got bigger and more complicated with a more number of diagrams with some good reasons. Although providing different diagrams for modeling specific parts of a software system, not all of them need to be applied in most cases. Therefore, heuristics, design guidelines, lessons learned from experiences are extremely important for the effective use of UML and to avoid unnecessary complication.

BP-UML'05 (Best Practices of the UML) is the first edition of this International Workshop held with the 24th International Conference on Conceptual Modeling (ER 2005). This workshop will be an international forum for exchanging ideas on the best and new practices of the UML in modeling and system developments. The workshop will be a forum for users, researchers, analyzers, and designers who use the UML to develop systems and software. To keep the high quality of former workshops held in conjunction with ER, a strong International Program Committee was organized with extensive experience in the UML and also taking into consideration its relevant scientific production in the area.

The workshop attracted papers from 13 different countries distributed over all continents such as The Netherlands, France, Spain, Israel, Korea, USA, Canada and Australia. We received 27 submissions and only 9 papers were selected by the Program Committee, making an acceptance rate of 36%.

The accepted papers were organized in three different sessions. In the first one, two papers will present valuable experience reports and another one will describe how to apply UML for multidimedia modeling. In the second one, one paper will be focused on evaluating the cardinality interpretation by users in a UML class diagram, and the other two papers will be focused on the Use case diagrams of the UML. Finally, in the third session, while one paper will present how to analyze the consistency of a UML diagram, the other two will be focused on the Model Driven Architecture (MDA) and metamodeling.

I would like to express my gratitude to the program committee members and the additional external referees for their hard work in reviewing papers, the authors for submitting their papers and the ER2005 organizing committee for all their support. This workshop was organized within the framework of the following projects: MESSENGER (PCC-03-003-2), METASIGN (TIN2004-00779) and DADASMECA (GV05/220). Thanks to the number of submissions of this first edition together with the high quality of the accepted papers, my intention is to organize the second edition of BP-UML again next year with ER2006.

J. Akoka et al. (Eds.): ER Workshops 2005, LNCS 3770, p. 1, 2005.

Improving Object-Oriented Micro Architectural Design Through Knowledge Systematization

Javier Garzás[1] and Mario Piattini[2]

[1] OCU, Oficina de Cooperación Universitaria, Chief Technology Officer, C / Arequipa, 1, Bloque I, Planta 5, Madrid, Spain
javierg@ocu.es, jgarzas@gmail.com
[2] Alarcos Research Group. University of Castilla-La Mancha, Ronda de Calatrava, s/n. 13071, Ciudad Real, Spain
Mario.Piattini@uclm.es

Abstract. Designers have accumulated much knowledge referring to OO systems design and construction, but this large body of knowledge is neither organized nor unified yet. In order to improve OO micro architectures, using the accumulated knowledge in a more systematic and effective way, we have defined a rules catalog (that unifies knowledge such as heuristics, principles, bad smells, etc.), the relationships between rules and patterns and an improvement method based on these subjects. We have carried out a controlled experiment which shows us that the usage of a rules catalog and its relationship with patterns really improves OO micro architectures.

1 Introduction

According to the [10], design is both "the process of defining the architecture, components, interfaces, and other characteristics of a system or components" and "the results of (that) process". A design must describe the architecture of a system, how the system is decomposed and organized into components. Generally, in software engineering the design is performed at two abstraction levels: macro architectural (high level) and micro architectural (low level). Object Oriented (OO) Micro architectural design is an old and well-known area within software engineering. Designers have accumulated a large body of knowledge regarding OO micro architectural design. Nevertheless, it is neither organized nor unified yet, and this area is still suffering from a lack of structured and classified knowledge.

In OO Micro-Architectural Design, ***patterns*** are the most popular and most refined example of accumulated knowledge; [4], [9], [3], [5], [6], etc, are popular references in this field. There are many and popular examples of design patterns, as *Observer, Decorator, State* or *Command* [9]. Nevertheless, even now, patterns application implicates several types of problems: difficult application, difficult learning, temptation to recast everything as a pattern, pattern overload, deficiencies in catalogs, and so on. Therefore, nowadays, patterns application is a real and important problem, and this fact has been brought up at several major congresses, for example OOPSLA 2001 - Workshop *"Beyond Design: Patterns (mis)used"*, where authors such as [20]

J. Akoka et al. (Eds.): ER Workshops 2005, LNCS 3770, pp. 444 – 453, 2005.

say *"We got more and more aware that a good description of the proposed solution is necessary, but useless for the reader if the problem and the forces that drive the relationship between problem and solution are not properly covered"*. Furthermore, several important books have dealt with this problem [8]. In many cases, the reason of these important problems is that the OO micro architectural design knowledge is associated exclusively with the pattern concept without taking into consideration that other elements of knowledge exist, such as ***principles, heuristics, best practices, bad smells***, etc.

According to ***principles,*** the main contributions are [14], [9] and [13]; and examples of principles are the *Dependency Inversion Principle, Don't Concrete Super class Principle*, etc. With regard to ***heuristics***, we can refer mainly to [19] or [2], for example: *"if two or more classes only share a common interface (i.e. messages, not methods), then they should inherit from a common base class only if they will be used polymorphically."* [19]. *Concerning* best practices, *we can highlight the Venners's work, for example:* "see objects as bundles of behaviour, not bundles of data." [22]. Regarding ***bad smells,*** the main work in this field is that of [7] in which several bad smells such as *"Refused request, Subclasses that do not use what they inherit" are enumerated*. But, again, the application of these bad smells and the differences between them are not clear: many of them concern a single concept with different names, while others sometimes do not contain knowledge gained from experience, and still others are simply vague concepts. This confusion leads to a less efficient use of knowledge.

In order to improve OO micro-architectural designs, using all the accumulated knowledge in a more systematic and effective way, we have defined a rules catalog (which unifies knowledge such as heuristics, principles, bad smells, etc.), and the relationship between rules and patterns. We have also created an improvement method for knowledge application which will be stated in the next section. In section three, we will summarize a controlled experiment aimed at demonstrating that the improvement method could really help us improve OO micro architectures. In section four, we will carry out a short presentation about the related work. Finally, the conclusion will point out the most important arguments of our work as well as the further research.

2 A Method to Use OOD Knowledge to Improve Micro Architectural Quality

We have observed that principles, heuristics, best practices, bad smells, etc., have an analogous structure, since all of them can be expressed as a Rule – they posit a condition and offer a recommendation. It should be stressed that the "recommendation" is not a solution like that of a pattern. Patterns are more formalized than Rules and pattern descriptions are always broader. They propose solutions to problems, while rules are recommendations which a design should fulfill. Unlike patterns, rules are highly based on using natural language, which can be more ambiguous [15].

In this regard, we have developed a unified rules catalog (some examples are shown in table 1), which are named and unified according to their condition in order to improve their detection. To describe these rules, we have used the sections used in [9] catalog to describe a design pattern (See table 2): Name, Intent, Also known as, Motivation, Structure, Applicability, Participants, Collaborations, Consequences, and Known Uses. With the exception of Implementation and Sample code (renamed and unified as Sample Design) and Related Patterns (renamed as "Implies the use of Patterns")

Table 1. Some OOD Rules

Rule of IF There are dependencies on concrete classes
Rule of IF an object has a different behavior according to its state
Rule of IF a class hierarchy is composed of many classes
Rule of IF anything is used a bit or never
Rule of IF a super class knows to any sub class
Rule of IF a class collaborates with many others
Rule of IF an interface changes to many clients
Rule of IF between an interface and its implementation there is not an abstract class
Rule of IF a super class is concrete
Rule of IF in a service there are many parameters
Rule of IF a class is very large
Rule of IF users interface elements are on domain class
Rule of IF a class use more external things than those of its own
Rule of IF a class refuses any delegate
Rule of IF the attributes classes are public or protected

The last section of the rule description reflects that rules could imply the use of patterns. Often, when we introduce a rule, we obtain a new design, which needs a pattern. One example of this situation is the application of *if there are dependencies on concrete classes* rule, which introduces an abstract class or an interface, which in turn needs a creational pattern [9] to create instances and associate objects in the new situation. We can observe that this does not always happen (cardinality 0 to n), not all the rules imply the introduction of a pattern; a clear example of this can be seen when we apply rules which work only inside a module, for example the "Long Method" Rule, a bad smell according to [7]. Therefore, for each rule in the catalog, if apply, we have enumerated the patterns implied by it, and these patterns are listed in the "Imply the use of (patterns)" section.

Based on this rule catalog, we have defined an OO micro architecture improvement method, which helps designers in their improvement activities (figure 1). In a first step, violated rules are detected within a micro architecture. This step is performed using the rules catalog. A second step consists of identifying the related patterns implicated by rules. With all this knowledge, we can improve the quality of OO micro architecture in a rational and systematized way.

Table 2. Detail of a Rule

NAME: "IF THERE ARE DEPENDENCIES ON CONCRETE CLASSES"

Intent

Strategy to depend on interfaces or abstract classes rather than on concrete elements.

Also known as

Dependency Inversion Principle [13] or Programming to an Interface, not an Implementation [9].

Motivation

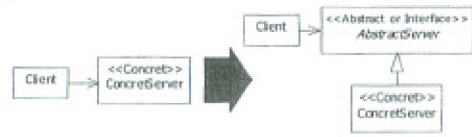
The structural design shows a particular type of dependency where high level modules depend on low-level ones. So, why do high-level modules depend directly on implementation modules? OO architecture shows a dependency mostly on abstractions and the modules containing implementation details also depend on these abstractions, not vice versa. The dependency has been inverted. This rule implies that each dependency within the design must have as its objective an interface or an abstract class; the dependencies must not have concrete classes as objectives. Concrete things are much more likely to change than abstract ones.

Applicability

Use this rule when: *you find dependencies on, or associations with concrete classes which may change.* Do not use this rule: *If dependency exists on a concrete class which is not likely to change (for example, a library class such as String).*

Recommendation

IF there are dependencies on concrete classes, **THEN** these dependencies should be on abstractions.



Participants

Client, Concrete Server (implementation) and Abstract Server Class (Concrete Server Interface).

Collaborations

Client communicates with Abstract Server Class and Concrete Server implements Abstract Server.

Consequences

Among others, this rule has the following benefits and liabilities: To introduce abstractions with which the design can be extended without being modified, to limit the impact of the variations in design, all the subclasses can respond to requests of the interface, and the subtypes of the abstract class and the clients will not be aware of the specific types of the objects being used.

Known Uses

This rule is used in many design patterns, frameworks, and components models.

Implies the use of (Patterns) One of the most common places where design depends on concrete classes is when instances are created. Concrete classes have to be instanced and the creational patterns (Abstract Factory, Builder, Factory Method, Prototype and Singleton) allow this instantiation.

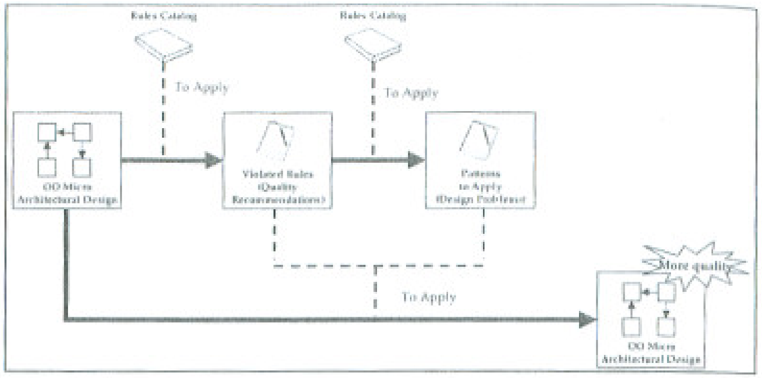


Fig. 1. A method to improve OO micro architectural quality

3 Empirical Validation

In this section, we will present a description of the main steps of the process followed to carry out the empirical validation, which is based on [23] [16] [17] [18] and [11] [12]. The main objective of this controlled experiment was to compare the effectiveness and efficiency of "traditional" OO design improvement methods with this new approach. Moreover, we aimed at analyzing if disposing of a rules catalog that unifies design knowledge as principles, best practices, heuristics, etc., and their relations with patterns has influence on the effectiveness and efficiency in the improving of the quality of OO micro architectures. Based on the GQM (Goal Question Metrics) template, the goal definition of our experiment can be summarized as follows:

Analyze	The improvement method based on the rules catalog
for the purpose of	evaluating
with respect to	effectiveness and efficiency
form the point of view of	software engineers
in the context of	software companies in Spain

3.1 Planning

The experiment is specific since it is focused on one technique applied to one domain; the ability to generalize from this specific context is further elaborated below when discussing threats to the experiment. The experiment addresses a real problem, i.e., if the method presented is more effective and efficient to be used in OO micro architectural quality improvement. Eighteen professionals of two companies carried out the experiment. The selected subjects were professionals having extensive experience in OO Design. We classified the subjects into two groups according to

their professional experience. The subjects were asked to fill a questionnaire out about their expertise, and taking into consideration the collected responses, we formed two groups of subjects, trying to have the same number of subjects with good marks and bad marks in each group. Both groups had [9] patterns catalog, but only one of them had the rules catalog. In addition to this, in a previous 30 minutes session, we explained to this group some notions about rules and their relationships with patterns; and how to apply the rules catalog. For each subject, we had prepared a folder with the experimental material. Each folder contained one micro architectural diagram and a questionnaire for answers.

We had to consider what independent variables or factors were likely to have an impact on the results. These are OO Micro Architecture. We considered two dependent variables [21]:

- **Effectiveness:** Number of Defects Found / Total Number of Defects. This is the percentage of the true improvements found by a designer with respect to the total number of defects.
- **Efficiency:** Number of Defects found / Inspection Time. Where Inspection Time is related to the time that subjects spent on inspecting the micro architecture; it is measured in minutes.

3.2 Hypotheses Formulation

Our purpose was to test two groups of hypotheses, one for each dependent variable.

Effectiveness Hypotheses

- H0,1. There is no difference regarding effectiveness of subjects in detecting the violation of rules using a rules catalog and their relationship with patterns as compared to subjects without using the rules catalog. // H1,1 : ¬ H0,1
- H0,2. There is no difference regarding effectiveness of subjects in detecting the application of patterns implicated by rules using a rules catalog s and their relationship with patterns as compared to subjects without using the rules catalog. . // H1,2 : ¬ H0,2
- $H_{0,3}$. There is no difference regarding effectiveness of subjects in detecting the application of patterns not implicated by rules using a rules catalog and their relationship with patterns as compared to subject without using the rules catalog // $H_{1,3}$: ¬ $H_{0,3}$

Efficiency Hypotheses

- H0,4. There is no difference regarding efficiency of subjects in detecting the violation of rules using a rules catalog and their relationship with patterns as compared to subjects without using the rules catalog. . // H1,4 : ¬ H0,4
- H0,5. There is no difference regarding efficiency of subjects in detecting the application of patterns implicated by rules using a rules catalog and their relationship with patterns as compared to subjects without using the rules catalog. . // H1,5 : ¬ H0,5
- H0,6. There is no difference regarding efficiency of subjects in detecting the application of patterns not implicated by rules using a rules catalog and their

relationship with patterns as compared to subjects without using the rules catalog. // H1,6 : ¬ H0,6

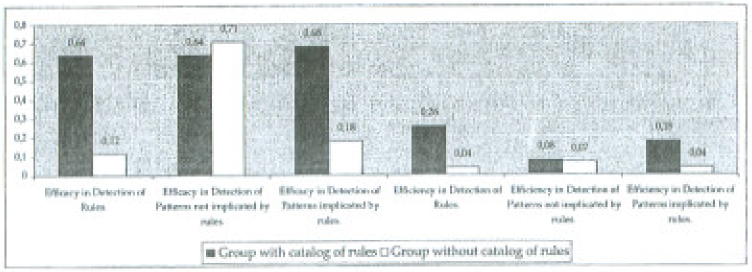


Fig. 2. Averages obtained from the experiment

3.3 Operation

In this section, we will describe the preparation, execution, and data validation of the experiment. Before the day of the experiment execution, we gave a seminar to the subjects of the group which would use the rules catalog. In this seminar, we explained to the subjects how to apply the rules catalog. The subjects had to fulfill manually their proposed solution, writing down the start and end time of the activity. We collected the forms filled out by the subjects, checking if they were complete.

3.4 Analysis and Interpretation

Figure 2 shows the averages obtained from the experiment. Outliers have not been identified. In order to decide how to test the validity of the hypotheses, we evaluated if the data followed a normal distribution, the result was normal, we decided to perform a t-Student test. In table 3, the results obtained by means of t-Student are shown. The first column represents the t-stat and the second column shows the t critical two – tail.

Table 3. Results obtained by means of t-Student

	t stat	t Critical two-tail
Efficacy in Detection of Rules.	5.38887	2.26215
Efficacy in Detection of Patterns not implicated by rules.	-0.22360	2.20098
Efficacy in Detection of Patterns implicated by rules.	3.36269	2.20098
Efficiency in Detection of Rules.	7.03868	2.26215
Efficiency in Detection of Patterns not implicated by rules	0.22269	2.26215
Efficiency in Detection of Patterns implicated by rules	4.35678	2.17881

We have obtained the following results. Firstly, it was confirmed by the t-Student test that the group with the rules catalog obtained better results in "Efficacy and Efficiency in Detection of Rules" and "Efficacy and Efficiency in Detection of Patterns implicated by rules". In the second place, the t-Student test could not confirm that the group with the rules catalog obtained better results in "Efficiency in Detection of Patterns not implicated by rules". However, this group obtained better averages; we have to highlight that "Efficiency in Detection n of Patterns not implicated by rules" is not influenced by rules catalog, since these patterns are not in catalog because they are not implicated by rules, and the application of these patterns will result in the detection of design problems more than design recommendations. Lastly, in a similar way, we could not confirm by using the t-Student test that the group without the rules catalog obtained better results in "Efficacy in Detection of Patterns not implicated by rules"; however, again, this result is not influenced by rules catalog.

3.5 Threats to Validity

A list of issues that threats the validity of the empirical study is identified below:

Conclusion Validity
The results confirmed by means of the t-Student test that there was a significant difference between the two groups, and that the new approach seems to be more effective and efficient for carrying out the OO micro architectural quality improvement. The statistical assumptions of each test were verified, so that, the conclusion validity was fulfilled.

Internal Validity

- Differences among subjects. We formed two groups, and the subjects were apportioned between these two groups according to their expertise and skills. For this reason, the subjects were asked to fill out a questionnaire about their expertise, and taking into account the collected responses, we formed the two groups of subjects.
- Differences among OOD diagrams. We used only one OOD diagram.
- Precision in the time values. The subjects were responsible for recording the start and finish times of each test, so they could introduce some imprecision but we think it is not very significant.
- Fatigue effects. The average time for carrying out the experiment was 20 minutes, so fatigue effects were minor.
- Persistence effects. Persistence effects are not present because the subjects had never participated in a similar experiment.
- Subject motivation. The subjects were very motivated.
- Other factors, such as plagiarism and influence among subjects were controlled.

External Validity

- Subjects. We are aware that more experiments with professionals must be carried out in order to be able to generalize these results. However, the subjects could be considered "common" OO designers at least in the context of Spanish software companies.

- Material used. We believe that the documents used might not be representative of an industrial problem, so more experiments with larger diagrams are needed.

5 Conclusions and Future Work

There has not been much effort made on empirical studies about OO design knowledge, and the few works we have found are mainly focused on design patterns. We should consider that (a) OO micro architectural design knowledge is associated with the pattern concept, but other elements exist, such as ***principles, heuristics, best practices, bad smells***, etc. (b) these other elements show a confused description, unification, definition, etc. (c) rules and patterns are related elements (d) with a unified rules catalog and related patterns is easier to detect quality recommendations and to apply patterns.

We are conscious that, in this experiment we have chosen to investigate with an individual technique that could interact with many other development techniques and procedures ("the life-cycle issue", [12]), the life-cycle model (light or weight) [1], the modeling language used to express the design artifacts, etc. So, large-scale empirical studies are needed to obtain conclusions about the effect of the knowledge systematization in the OOD improvement.

References

1. Boehm B. and Turner R. (2003). Using Risk to Balance Agile and Plan-Driven Methods. Computer, Vol. 36, No. 6, pp. 57-66.
2. Booch G. (1996). Managing the Object-Oriented project. Addison-Wesley
3. Buschmann F., Meunier R., Rohnert H., Sommerlad P. and Stal M. (1996). A System of Patterns: Pattern-Oriented Software Architecture, Addison-Wesley.
4. Coad P. (1992). Object-Oriented Patterns. *Comunications ACM*, vol. 35, n 9, pp. 152-159.
5. Coplien J. and Schmidt. (1996). Patterns Languages of Program Design. Addison Wesley.
6. Tichy W. (1997). A catalogue of general-purpose software design patterns. TOOLS '97.
7. Fowler M. (2000). Refactoring improving the design of existing code. Addison Wesley
8. Fowler M. (2002). Patterns of Enterprise Application Architecture. Addison Wesley.
9. Gamma E., Helm R., Johnson R. and Vlissides J. (1995). *Design patterns: Elements of Reusable Object Oriented Software.* Addison-Wesley.
10. IEEE (1990). STD 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology.
11. Kitchenham B. A., Pfleeger S., Pickard L., Jones P., Hoaglin D., El Emam K., and Rosenberg J. (2002). Preliminary Guidelines for Empirical Research in Software Engineering. IEEE Transactions on Software Engineering, IEEE Computer Society, 28 (8).
12. Kitchenham, B.A., Dybå, T. and Jorgensen, M. (2004). Evidence-based Software Engineering. Procedigns of the 26th International Conference on Software Engineering (ICSE'04), IEEE.
13. Martin 1996. The Dependency Inversion Principle. C++ Report.
14. Meyer B. (1988). *Object Oriented Software Construction.* Prentice Hall.

15. Pescio C. (1997). Principles Versus Patterns. IEEE Computer, 30 (9): 130-131.
16. Prechelt L., Unger B., Philippsen M., and Tichy W. (1997). Two Controlled Experiments Assessing the Usefulness of Design Pattern Information During Program Maintenance. IEEE Transactions on Software Engineering, 28(6), pp. 595-606.
17. Prechelt L. and Unger B. (1998). A Series of Controlled Experiments on Design Patterns: Methodology and Results. Proc. Softwaretechnik '98 GI Conference (Softwaretechnik-Trends 18(3)), pp. 53-60, Paderborn, September 7-9, 1998.
18. Prechelt L., Unger B., Tichy W., Bössler P. and Votta G. (2001). A Controlled Experiments in Maintenance Comparing Design Patterns to Simpler Solutions. IEEE Transactions on Software Engineering. 27(12), pp. 1134-1144.
19. Riel A. J. (1996). *Object-Oriented Design Heuristics.* Addison-Wesley.
20. Schwanninger C. (2001). "Patterns as problem indicators" Workshop on Beyond Design Patterns (mis)Used, OOPSLA 2001.
21. Thelin T., Runeson P., Wholin C., Olsson T. and Andersson C. (2004). Evaluation of Usage – Based Reading – Conclusions after three Experiments. Empirical Software Engineering, 9 (1-2): 77 - 110.
22. Venners B. (2004). *Interface Design Best Practices in Object-Oriented API Design in Java.* Retrieved February 2004 from http://www.artima.com/interfacedesign/contents.html
23. Wohlin C., Runeson P., Höst M., C. Ohlsson M., Regnell B., and Wesslén A. (2000). Experimentation in Software Engineering – An Introduction. Kluwer Academic Publishers.