ʒes in Computer Science

eries reports state-of-the-art results in computer science
velopment, and education, at a high level and in both printed
ic form. Enjoying tight cooperation with the R&D community,
ous individuals, as well as with prestigious organizations and
ICS has grown into the most comprehensive computer science
um available.

t LNCS, including its subseries LNAI, spans the whole range of
ience and information technology including interdisciplinary
ariety of application fields. The type of material published
includes

gs (published in time for the respective conference)

eedings (consisting of thoroughly revised final full papers)

nonographs (which may be based on outstanding PhD work,
rojects, technical reports, etc.)

ly, several color-cover sublines have been added featuring,
llection of papers, various added-value components; these
lude

textbook-like monographs or collections of lectures given at
courses)

e-art surveys (offering complete and mediated coverage

(introducing emergent topics to the broader community)

ɔ the printed book, each new volume is published
y in LNCS Online at *http://www.springerlink.com/series/lncs/*.

ɔrmation on LNCS can be found at the series home page
pringer.de/comp/lncs/.

r publication should be sent to

cial, Tiergartenstr. 17, 69121 Heidelberg, Germany

;@springer.de

ʼ743

Lecture Notes in Computer Science

LNCS LNAI LNBI

Olivé · Yoshikawa
Yu et al. (Eds.)

Advanced Conceptual

2002
ɛshops

LNCS 2784

Antoni Olivé
Masatoshi Yoshikawa
Eric S.K. Yu et al. (Eds.)

Advanced Conceptual Modeling Techniques

ER 2002 Workshops – ECDM, MobIMod, IWCMQ, and eCOMO
Tampere, Finland, October 2002
Revised Papers

Springer

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Marcela Genero Fabio Grandi
Willem-Jan van den Heuvel John Krogstie
Kalle Lyytinen Heinrich C. Mayr
Jim Nelson Antoni Olivé Mario Piattini
Geert Poels John Roddick Keng Siau
Masatoshi Yoshikawa Eric S.K. Yu (Eds.)

Advanced Conceptual Modeling Techniques

ER 2002 Workshops
ECDM, MobIMod, IWCMQ, and eCOMO
Tampere, Finland, October 7-11, 2002
Revised Papers

Springer

- Change Detection, Monitoring and Mining
- Evolution and Change in Internet-Based Information Systems
- Evolution and Change in E-services and E-world Systems
- Induction of Cause and Effect, Logics for Evolution
- Maintenance of Views, Summaries, Dictionaries and Warehouses
- Managing Evolution of Sources in Information Integration

With respect to the main ER conference, the ECDM workshop aims at stressing the evolutionary aspects involved in conceptual modelling and in the development and implementation of systems, ranging from the modelling of information dynamics to the dynamics of the modelling process itself. Another explicit aim of ECDM 2002 (as it was also for ECDM 1999) was to bring together scientists and pratictioners interested in evolution and change aspects in different research fields and, thus, people who often belong to completely separate communities. It is our opinion that such interactions can be tighter and cross-fertilization can be more useful in the context of a collaborative workshop like ECDM than in the context of the main conference sessions. Moreover, since the emphasis is on the evolutionary dimension, a special insight is sought upon this specific aspect, one that could hardly find an appropriately broad coverage in the scope of the main ER conference.

Following the acceptance of the workshop proposal by the ER 2002 organizing committee, an international and highly qualified program committee was assembled from research centers worldwide. As a result of the call for papers, the program committee received 19 submissions from 15 countries, and after rigorous refereeing 10 high-quality papers were eventually chosen for presentation at the workshop, and these appear in these proceedings.

We would like to thank both the program committee members and the additional external referees for their timely expertise in reviewing the papers. We would also like to thank all authors for submitting their papers to this workshop. Last, but not least, we would like to thank the ER 2002 organizers for their support, and in particular the workshop co-chairs, Antoni Olivé, Eric Yu, and Masatoshi Yoshikawa.

September 2002

Fabio Grandi and John Roddick
Program Co-chairs
ECDM 2002
http://kdm.first.flinders.edu.au/events/ECDM02.html

MobIMod 2002

Mobility is perhaps the most important market and technological trend in information and communication technology. With the advent of new mobile infrastructures providing higher bandwidths and constant connections to the network from virtually everywhere, the way people use information resources is predicted to be radically transformed. The rapid developments in information technology (IT) are substantially changing the landscape of organizational computing. Workers in many business areas are becoming increasingly mobile. Workers in more and more areas will be required to act more flexibly within the constraints of the business processes they are currently engaged in. At the same time they will often want to use the same information technology to support their private tasks. During the last few years, a new breed of information system has emerged to address this situation, referred to as m-commerce systems or mobile information systems. The objective of the workshop was to provide a forum for researchers and practitioners interested in modeling methods for mobile information systems to meet, and exchange research ideas and results.

The relevant topics for the workshop included the following aspects of m-commerce and mobile information systems:

- Mobile commerce models and architecture
- Service modeling
- Mobile access to enterprise systems (ERP, CRM, SCM, etc.)
- Enterprise modeling and business process re-engineering
- Workflow modeling
- Meta-modeling and method engineering
- Evaluation of modeling languages and experience
- Modeling of access control to provide security and privacy
- Content personalization and user modeling
- Context modeling
- Requirement modeling
- Information and database modeling
- Component engineering and integration
- Geographical information systems and location-based services
- Cross-platform conceptual interface modeling
- Mobile modeling tools
- Modeling of embedded systems
- (Mobile) Agent modeling and design
- Agile modeling, extreme modeling, and extreme programming

October 2002

John Krogstie
Program Chair
MobIMod 2002

IWCMQ 2002

Conceptual modeling has been recognized as a key task that lays the foundation of all later design and implementation work. The early focus on conceptual modeling may help in building better systems, without unnecessary rework at later stages of the development when changes are more expensive and more difficult

Organization

ER 2002 Workshops Chairs

Antoni Olivé	Universitat Polytecnica de Catalunya, Spain
Masatoshi Yoshikawa	Nagoya University, Japan
Eric S.K. Yu	Univeristy of Toronto, Canada

ECDM 2002 Program Committee Co-chairs

Fabio Grandi	University of Bologna, Italy
John Roddick	Flinders University, South Australia

ECDM 2002 Program Committee

Michael Böhlen	Aalborg University, Denmark
Curtis Dyreson	Washington State University, USA
Ramez Elmasri	University of Texas, USA
Enrico Franconi	University of Manchester, UK
Kathleen Hornsby	University of Maine, USA
Richard McClatchey	University of the West of England, UK
Erik Proper	University of Nijmegen, The Netherlands
Chris Rainsford	DSTO, Australia
Sudha Ram	University of Arizona, USA
Elke Rundensteiner	Worcester Polytechnic Institute, USA
Maria Rita Scalas	University of Bologna, Italy
Myra Spiliopoulou	Leipzig Graduate School of Management, Germany
Babis Theodoulidis	UMIST, UK
Carlo Zaniolo	UCLA, USA

MobIMod 2002 Workshop Co-chairs

John Krogstie
Kalle Lyytinen
Keng Siau

MobIMod 2002 Program Committee

R. Baskerville	J. Dietz	R. Kalakota
S. Brinkkemper	G. Grosz	S. Kelly
J. Dhaliwal	H. Kangassalo	J. Krogstie

Table of Contents

Second International Workshop on Evolution and Change in Data Management (ECDM 2002)

Management of Time and Changes in Information Systems

Architectures, Models, and Tools for Systems Evolution

Change Management for a Temporal Versioned Object-Oriented Database*

Renata de Matos Galante, Nina Edelweiss, and Clesio Saraiva dos Santos

Instituto de Informática - Universidade Federal do Rio Grande do Sul
{galante,nina,clesio}@inf.ufrgs.br

Abstract. In this paper, we propose a schema versioning mechanism to manage the schema evolution in temporal object-oriented databases. The schema evolution management uses an object-oriented data model that supports temporal features and versions definition - the Temporal Versions Model - TVM. One interesting feature of our proposal is that TVM is used to control not only the schema versioning, but also the storage of extensional database and propagation of the changes performed on the objects. The extensional data level supports integration with the existing database, allowing the maintenance of conventional and temporal versioned objects. The instance propagation approach is proposed through the specification of propagation and conversion functions. These functions assure the correct instance propagation and allow the user to handle all instances consistently in both backward and forward schema versions. Finally, the initial requirements concerning data management in the temporal versioning environment, during schema evolution, are presented.

1 Introduction

Object-oriented databases offer powerful modeling concepts as those required by advanced application domains as CAD and Case tools. Typical applications handle large and complex structured objects, which frequently change their value and structure. As the structure is described in the database schema, support to schema evolution is a highly required feature. In this context, the version concept has been applied to maintain all the history of the database evolution.

Schema evolution and schema versioning are two techniques that allow schema modifications while consistency is maintained between a schema and its data. According to accepted terminology [1], a database supports schema evolution if it allows schema changes without losing extentional data. In addition, the schema versioning support allows not only the maintenance of data, but also the access to all data through schema versions.

However, the representation of the temporal dimension is essential to keep the whole evolution history. This feature is necessary in many computer applications, as medical control, geographical information systems and flight reservation. Schema versioning with temporal features has been studied extensively

* This work has been partially supported by Capes and CNPq.

A. Olivé et al. (Eds.): ER 2002 Ws, LNCS 2784, pp. 1–12, 2003.

Assessing Object-Oriented Conceptual Models Maintainability

Marcela Genero, José Olivas, Mario Piattini, and Francisco Romero

Department of Computer Science, University of Castilla-La Mancha.
Paseo de la Universidad, 4, 13071, Ciudad Real (Spain)
{marcela.genero,joseangel.olivas,mario.piattini}@uclm.es

Abstract. Conceptual modeling has become a key task in the early phases of object oriented (OO) software life cycle. In the development of OO software class diagrams represent the conceptual schema that reflects not only the objects of the application domain but also the behaviour of them. Indeed, class diagrams constitute the backbone of OO software so, their quality has a great impact on the quality of the product which is finally implemented. To assess class diagram quality, it is useful to have quantitative and objective measurement instruments. After having thoroughly reviewed existing OO measures applicable to class diagrams at a high-level design stage, we defined a set of metrics for UML class diagram structural complexity (and internal quality attribute), with the idea that it is related to maintainability of such diagrams. In order to gather empirical evidence that the proposed metrics could be early indicators of class diagrams maintainability, we carried out a controlled experiment. The main goal of this paper is to show each of the steps of the experimental process, and how we have built a prediction model for class diagram maintainability based upon the data collected in the experiment using a novel process, the Fuzzy Prototypical Knowledge Discovery process.

1 Introduction

Conceptual modeling has become a key task in the early phases of OO software life cycle. The proof of this is that modern approaches towards OO system development, like Catalysis [12] and Rational Unified Process [30], includes conceptual modelling as a relevant task. In the development of OO software, class diagrams represent the conceptual schema that reflects not only the objects of the application domain but also the behaviour of them. Indeed, the class diagram is a key early artifact that lays the foundation of all later design and implementation work. Hence, class diagram quality is a crucial issue that must be evaluated (and improved if necessary) in order to obtain quality OO software, which is the main concern of present day software development organisations.

It is in this arena where software measurement plays an important role, because the early availability of metrics contributes to class diagram quality evaluation in an objective way avoiding bias in the quality evaluation process. Moreover, metrics

A. Olivé (Eds.): ER 2003 Ws, LNCS 2784, pp. 288-299, 2003.

provide a valuable and objective insight into specific ways of enhancing each of the software quality characteristics.

Given that maintenance was (and will continue to be) the major resource consumer of the whole software life cycle, maintainability has become one of the software product quality characteristics [20] that software development organisations are more concerned about. However, we are aware that maintainability is an external quality attribute that can only be measured late in the OO software life cycle. Therefore, it is necessary to have early indicators of such qualities based, for example, on the structural properties of class diagrams [4].

After a thorough review of some of the existing OO measures, applicable to class diagrams at conceptual level [6, 10, 22, 23] we have proposed [16, 18] a set of UML class diagram structural complexity measures brought on by the use of UML relationships, such as associations, generalizations, aggregations and dependencies (see table 1).

Table 1. Metrics for UML class diagram structural complexity

Metric name	Metric definition
Number of Classes (NC)	The total number of classes.
Number of Attributes (NA)	The total number of attributes.
Number of Methods (NM)	The total number of methods .
Number of Associations (NAssoc)	The total number of associations.
Number of Aggregation (NAgg)	The total number of aggregation relationships within a class diagram (each whole-part pair in an aggregation relationship).
Number of Dependencies (NDep)	The total number of dependency relationships.
Number of Generalisations (NGen)	The total number of generalisation relationships within a class diagram (each parent-child pair in a generalisation relationship).
Number of Aggregations hierarchies (NAggH)	The total number of aggregation hierarchies (whole-part structures) within a class diagram.
Number of Generalisations hierarchies (NGenH)	The total number of generalisation hierarchies within a class diagram.
Maximum DIT	It is the maximum of the DIT (Depth of Inheritance Tree) values obtained for each class of the class diagram. The DIT value for a class within a generalisation hierarchy is the longest path from the class to the root of the hierarchy.
Maximum Hagg	It is the maximum of the HAgg values obtained for each class of the class diagram. The HAgg value for a class within an aggregation hierarchy is the longest path from the class to the leaves.

However, the proposal of metrics is of no value if their practical use is not demonstrated empirically, either by means of case studies taken from real projects or by controlled experiments. Empirical validation is crucial for the success of any software measurement project [2, 15, 21, 31]. Therefore, our main motivation is to the start and end time. The difference between the two is what we call maintenance time (expressed in minutes and seconds). The modifications to each class diagram were similar, including adding, updating and deleting attributes, methods, classes, associations, etc.

The subjects were given all the materials described in the previous paragraph. We explained how to do the tests. We allowed one week to carry out the experiment, i.e., each subject had to do the test alone, and could use unlimited time to solve it.

We collected all the data including the modified class diagrams with the maintenance time obtained from the responses of the tests and the metrics values automatically calculated by means of a metric tool we designed.

Once the data was collected, we controlled if the tests were complete and if the modifications had been done correctly. We discarded the tests of seven subjects, which included a required modification that was done incorrectly. Therefore, we took into account the responses of 23 subjects.

2.1 Analysis and Interpretation

We had the metric values calculated for each class diagram (see table 5), and we calculated the mean of the maintenance time. So this is the data we want to analyse to test the hypotheses stated above. We applied the Kolmogrov-Smirnov test to ascertain if the distribution of the data collected was normal. As the data were non-normal we decided to use a non-parametric test like Spearman's correlation coefficient, with a level of significance $\alpha = 0.05$, correlating each of the metrics separately with maintenance time (see table 2).

Table 2. Spearman´s correlation coefficients between metrics and maintenance time

Metrics	Spearman´correlation coefficients
NC	0.941 p=0
NA	0.803 p=0.009
NM	0.795 p=0.01
NAssoc	0.671 p=0.006
NAgg	0.667 p=0.049
NDep	0.411 p=0.272
NGen	0.728 p=0.04
NAggH	0.759 p=0.018
NGenH	0.719 p=0.029
MaxHAgg	0.840 p=0.005
MaxDIT	0.669 p=0.04

For a sample size of 9 (mean values for each diagram) and $\alpha = 0.05$, the Spearman cutoff for accepting H_0 is 0.66 [5, 11]. Because the computed Spearman's correlation coefficients (see table 6) for all the metrics, except for NDep, are above the cutoff, and the p-value < 0,05, the null hypothesis H_0, is rejected . Hence, we can conclude that there is a significant correlation between all the metrics (except NDep) and the maintenance time. So, NDep is the only one that has a no correlation, but this could

be explained by the fact that in most of the selected diagrams NDep took the value 0. So in future experiments we have to select diagrams with more representative NDep metric values.

2.2 Validity Evaluation

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

- **Threats to conclusion validity.** The conclusion validity defines the extent to which conclusions are statistically valid. The only issue that could affect the statistical validity of this study is the size of the sample data (243 values, 9 diagrams and 23 subjects). We are aware of this, so we will consider the results of the experiment as preliminary findings.
- **Threats to construct validity.** The construct validity is the degree to which the independent and the dependent variables are accurately measured by the measurement instruments used in the study. The dependent variable we used is maintenance time, i.e., the time each subject spent performing the tasks related to the modifications arising from the new requirements, so we consider this variable constructively valid. The construct validity of the measures used for the independent variables is guaranteed by Poels and Dedene´s framework [28] used for their theoretical validation [16].
- **Threats to internal validity.** The internal validity defines the degree of confidence in a cause-effect relationship between factors of interest and the observed results. Seeing the results of the experiment we can conclude that empirical evidence of the existing relationship between the independent and the dependent variables exists. We have tackled different aspects that could threaten the internal validity of the study, such as: differences among subjects, knowledge of the universe of discourse among class diagrams, precision in the time values, learning effects, fatigue effects, persistence effects and subject motivation.
- **Threats to External Validity.** External validity is the degree to which the research results can be generalised to the population under study (UML diagrams used as design artifacts for developing OO software) and to other research settings. The greater the external validity, the more the results of an empirical study can be generalised to actual software engineering practice. Two threats to validity have been identified which limit the ability to apply any such generalisation, and we tried to alleviate them: materials and tasks and subject selection.

Hereafter, we will show how we have applied our prediction model. Given a new class diagram, if you want to predict its maintainability there are two possibilities:

1. Evaluate which prototype has more affinity with, the new class diagram and, give as a result the maintenance time of a new class diagram the values taken from table 2. This is very trivial, and there is lose of information.
2. Using Fuzzy Deformable Prototypes [26] we can deform the most similar prototype to a new class diagram, and define the factors for a new situation, using a linear combination with the degrees of membership as coefficients. This solution is better than the previous because it adapts the prototype instead of basing it on fixed values. It also takes into account the degree of membership with other prototypes, without loosing valuable information.

We will show an example of how to deform the fuzzy prototypes found above. Given the metric values corresponding to a new class diagram shown in table 4 and normalising their values, the final average is 0.79.

Table 4. Metric values for a new class diagram

NC	NA	NM	NAssoc	NAgg	NDep	NGen	NAggH	NGenH	MaxDIT
21	30	70	10	6	3	16	2	3	2

The most similar prototype for this new class diagram is "High time-consuming to maintain", with a degree of membership of 0.81. The predicted values for the maintenance time related to the new class diagram is shown in table.

Table 5. Predicted maintanence time for a new class diagram

Average time	Minimum time	Maximum time
6 minutes 15 seconds	2 minutes	15 minutes 10 seconds

4 Conclusions and Future Work

Due to the growing demand of quality OO software, continuous attention to and assessment of class diagrams is necessary to produce quality OO software. As in the OO software development field it is generally accepted that the quality of software is highly dependent on decisions made early in the development, it is necessary to have measurement support for class diagrams early in the development life cycle in order to contribute to the quality of the OO software which is finally delivered.

In this paper we have presented a set of metrics for assessing the structural complexity of UML class diagrams, obtained at early phases of the OO software life cycle.

We have also carried out a controlled experiment, corroborating by means of it that there seems to be high correlation between the proposed metrics and the maintenance time. We have also shown how to predict UML class diagram maintainability based on the metrics values and the time spent on maintenance tasks, by a prediction model built using the FPKD process. Even though the FPKD has been used in other domains such forest fire prediction, medical diagnosis, etc. [25][26], we are aware that it is pending for future work the validation of the model presented in this paper.

Nevertheless, despite the encouraging results obtained more metric validation is needed in order to assess if the presented metrics could be really used as early quality indicators. Also, data of "real projects" on UML class diagram maintainability efforts would be useful, as well as time spent on maintenance tasks in order to predict data that can be highly fruitful to software designers and developers. However the scarcity of such data continues to be a great problem we must fin other ways to tackle validating metrics. Brito e Abreu et al. [7, 8, 9] suggested the necessity of a public repository of measurement experiences, which we think would be a good step towards the success of all the work done on software measurement.

In future work, we will focus our research on measuring other quality factors like those proposed in the ISO 9126 [20], which not only tackles static diagrams, such as class diagrams, but also evaluates dynamic UML diagrams, such as, statechart diagrams, activity diagrams, etc. To our knowledge, little work has been done towards measuring dynamic UML diagrams [13, 29, 33].

Acknowledgements

This research is part of the DOLMEN project supported by CICYT (TIC 2000-1673-C06-06) and the CIPRESES project supported by CICYT (TIC 2000-1362-C02-02).

We want to thanks Geert Poels for his valuable and encouraging comments while we are writing this work.

References

1. Basili, V., Rombach, H.: The TAME project: towards improvement-oriented software environments, IEEE Transactions on Software Engineering, Vol. 14 N° 6 , (1988) 728-738
2. Basili, V., Shull, F., Lanubile, F.: Building Knowledge through Families of Experiments. IEEE Transactions on Software Engineering, Vol. 25 N° 4, (1999) 435-437
3. Bell, R.: Analytic Issues in the Use of Repertory Grid Technique. Advances in Personal Construct Psychology 1, (1990) 25-48
4. Briand, L., Arisholm, S., Counsell, F., Houdek, F., Thévenod-Fosse, P.: Empirical Studies of Object-Oriented Artifacts, Methods, and Processes: State of the Art and Future Directions. Empirical Software Engineering, Vol. 4 N° 4, (1999) 387-404
5. Briand, L., El Emam, K., Morasca, S.: Theoretical and empirical validation of software product measures. Technical Report ISERN-95-03, International Software Engineering Research Network, (1995)
6. Brito e Abreu, F., Carapuça, R.: Object-Oriented Software Engineering: Measuring and controlling the development process. 4th International Conference on Software Quality, Mc Lean, Va, USA, (1994)